

## Sistema de Recolha de Garrafas Plástico

**PEDRO DANIEL DA COSTA PACHECO**

novembro de 2017

# SISTEMA DE RECOLHA DE GARRAFAS PLÁSTICO

Pedro Daniel Costa Pacheco



Departamento de Engenharia Electrotécnica  
Mestrado em Engenharia Eletrotécnica e de Computadores  
Área de Especialização em Automação e Sistemas

2017



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Pedro Daniel Costa Pacheco, Nº 1100460, 1100460@isep.ipp.pt

Orientação científica: Lino Manuel Baptista Figueiredo, lbf@isep.ipp.pt



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

10 de novembro de 2017



## *Agradecimentos*

As perspectivas de formação em contexto de pesquisa e revisão da literatura fomentam a reflexão potencializando o desenvolvimento formativo. Neste sentido, os diferentes atores do processo formativo adquirem especial relevo, tornando-se imprescindível salientar o seu contributo nesta etapa:

- Ao orientador de Tese, Engenheiro Lino Figueiredo um agradecimento especial pela disponibilidade, apoio e contributos fundamentais no percurso traçado;
- À minha família e namorada por todo o apoio proporcionado nesta etapa do meu processo formativo;
- A todos os meus amigos pelo companheirismo e amizade;

A todos um sincero Obrigado!

Esta página foi intencionalmente deixada em branco.

## *Resumo*

A carência no domínio da consciencialização social no âmbito da reciclagem e reaproveitamento dos materiais assume-se como potenciadora da limitação de recursos existentes. Neste sentido, e visando contrariar a realidade expressa, a conjugação da tecnologia com finalidades sociais assume o tema central da presente tese de mestrado. Evidenciada pelo desenvolvimento de um protótipo assente nas *Reverse Vending Machine*, a mesma visa ser capaz de responder às exigências económicas, sociais e ambientais, no que respeita à recolha de matérias plásticas e consequente retribuição ao consumidor. Versada nas tecnologias existentes e na revisão da literatura serão apresentados os domínios de hardware e software, neste último, evidenciando a análise de imagem assente nas características de Haar e respeitando o algoritmo de AdaBoost e cascata de Classificadores bem como o seu treino. Este, visa a recolha de garrafas de plásticas introduzidas pelo utilizador após o processo de análise de um símbolo, selecionado numa fase precedente, e consequente retribuição do valor monetário da garrafa ao utilizador.

## *Palavras-Chave*

*Reverse Vending Machine*, Reciclagem, Análise de Imagem em Tempo Real





## *Abstract*

The lack of social awareness in the recycling area and reuse of materials is seen as a potentiator of the existing resources limitation. In this sense, and in order to counteract the expressed reality, the conjugation of technology for social purposes takes the central theme of this master thesis. Evidenced by the development of a prototype based on the Reverse Vending Machine, it aims to be able to respond to the economic, social and environmental requirements, regarding the collection of plastic materials and consequent retribution to the consumer. Based on the existing technologies and the literature review, the hardware and software domains will be presented, showing the image analysis based on Haar characteristics and respecting the AdaBoost algorithm and cascade of Classifiers as well as their training. This prototype collects plastic bottles introduced by the user, after the process of analyzing a symbol, and consequently retribution of the monetary value defined for the bottle to the user.

## ***Key-Words***

*Reverse Vending Machine*, Recycling, Real-Time Image Analysis

Esta página foi intencionalmente deixada em branco.

# Índice

<b>AGRADECIMENTOS .....</b>	<b>I</b>
<b>RESUMO .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>V</b>
<b>ÍNDICE .....</b>	<b>VII</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>IX</b>
<b>ÍNDICE DE TABELAS .....</b>	<b>XI</b>
<b>ACRÓNIMOS.....</b>	<b>XIII</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. JUSTIFICAÇÃO E APRESENTAÇÃO DO TEMA .....	2
1.2. OBJETIVOS .....	2
1.3. ORGANIZAÇÃO DO RELATÓRIO .....	3
<b>2. ESTADO DA ARTE .....</b>	<b>5</b>
2.1. REVERSE VENDING MACHINE – RVM .....	5
2.2. ATUALIDADE: TECNOLOGIAS IMPLEMENTADAS .....	6
2.3. TECNOLOGIAS IMPLEMENTADAS NA REALIZAÇÃO E DESENVOLVIMENTO DO PROJETO .....	10
2.4. RASPBERRY PI .....	12
2.5. LINGUAGENS E BIBLIOTECAS DE PROGRAMAÇÃO .....	13
2.6. ANÁLISE DE IMAGEM .....	17
<b>3. PROJETO .....</b>	<b>27</b>
3.1. REQUISITOS .....	27
3.2. ARQUITETURA GERAL .....	33
3.3. DESCRIÇÃO DA TECNOLOGIA UTILIZADA .....	35
3.4. DESCRIÇÃO DE SOFTWARE .....	45
<b>4. IMPLEMENTAÇÃO .....</b>	<b>47</b>
4.1. IMPLEMENTAÇÃO DE HARDWARE .....	47
4.2. IMPLEMENTAÇÃO DE SOFTWARE .....	58
<b>5. TESTES E RESULTADOS.....</b>	<b>73</b>
<b>6. CONCLUSÃO.....</b>	<b>77</b>
<b>ANEXO A. DOWNLOAD IMAGENS NEGATIVAS.....</b>	<b>81</b>
<b>ANEXO B. FILTRAGEM IMAGENS NEGATIVAS.....</b>	<b>82</b>
<b>ANEXO C. CRIAÇÃO DO FICHEIRO BG.TXT .....</b>	<b>83</b>

<b>ANEXO D. TEMPO DE ANÁLISE DE IMAGEM .....</b>	<b>84</b>
<b>ANEXO E. PROGRAMA PRINCIPAL.....</b>	<b>85</b>
<b>ANEXO F. AMBIENTE GRÁFICO .....</b>	<b>92</b>

## *Índice de Figuras*

Figura 1 – Reverse Vending Machine .....	6
Figura 2 – Flow Technology .....	7
Figura 3 – Sure Return .....	8
Figura 4 – True Vision .....	9
Figura 5 – Sensor CMOS .....	11
Figura 6 – Filtro Bayer .....	11
Figura 7 – Raspberry Pi 2 .....	12
Figura 8 – Características Haar Retângulos .....	18
Figura 9 – Imagem Integral .....	19
Figura 10 – Cálculo Imagem Integral (imagem adaptada de [16]). .....	20
Figura 11 – Características Haar aplicadas a uma imagem [16] .....	23
Figura 12 – Cascata de Classificadores .....	24
Figura 13 – Plano de rotação do símbolo .....	25
Figura 14 – Modelo 3D Protótipo Mecânico .....	28
Figura 15 – Distância Camara ao Objeto .....	29
Figura 16 – Cálculo da Velocidade dos Rolos .....	32
Figura 17 – Arquitetura Geral do Sistema .....	34
Figura 18 – Arquitetura Geral do Sistema – Sistema de Recolha .....	35
Figura 19 – Conector DSI .....	37
Figura 20 – Detetor e Emissor infravermelho .....	39
Figura 21 – Optoacopladores .....	41
Figura 22 – Ponte H .....	42
Figura 23 – Ponte H: Exemplo I .....	43
Figura 24 – Ponte H Exemplo II .....	43
Figura 25 – Ponte H com Díodos .....	44
Figura 26 - Emissor de Infravermelhos .....	48
Figura 27 – Detetor de Infravermelhos .....	49
Figura 28 – Cálculo LEDs Alto Brilho .....	50
Figura 29 – Esquema Elétrico Geral .....	53
Figura 30 – PCB Geral Face Superior .....	53
Figura 31 – PCB Geral Face Inferior .....	54
Figura 32 – Esquema Elétrico do Detetor Infravermelho .....	54
Figura 33 – PCB Detetor Infravermelho .....	55
Figura 34 – Esquema Elétrico Emissor Infravermelho .....	55

Figura 35 – PCB Emissor Infravermelho .....	55
Figura 36 – Esquema Elétrico LEDs Alto Brilho.....	56
Figura 37 – PCB LEDs Alto Brilho .....	57
Figura 38 – Protótipo de testes .....	57
Figura 39 – Qt Creator IDE.....	59
Figura 40 – Símbolo a detetar .....	60
Figura 41 – GUI Desenvolvida em Kivy.....	62
Figura 42 – Ambiente Gráfico Página 0 – Inicial.....	63
Figura 43 – Ambiente Gráfico Página 1 – Recolha.....	63
Figura 44 – Ambiente Gráfico Página 2 – Contabilização de Garrafas .....	64
Figura 45 – Ambiente Gráfico Página 3 – Valor Retorno.....	64
Figura 46 – Ambiente Gráfico Página 4 – Impressão .....	65
Figura 47 – Ambiente Gráfico Página 6 – Valor Retorno Cheio .....	66
Figura 48 – Ambiente Gráfico Página 5 – Manutenção .....	66
Figura 49 – Fluxograma Ambiente Gráfico .....	67
Figura 50 – Fluxograma Geral .....	68
Figura 51 – Fluxograma Ciclo Processo .....	69
Figura 52 – Fluxograma Ciclo GUI .....	71

## *Índice de Tabelas*

Tabela 1	Regulador de Tensão .....	40
Tabela 2	Treinos Realizados .....	73
Tabela 3	Tempo de treinamento.....	74
Tabela 4	Treino Selecionado.....	75





## *Acrónimos*

ANSI	–	<i>American National Standard Institute</i>
CI	–	<i>Circuito Integrado</i>
CNC	–	<i>Computer Numerical Control</i>
CSI	–	<i>Camera Serial Interface</i>
DC	–	<i>Direct Current</i>
DCS	–	<i>Display Command Set</i>
DSI	–	<i>Display Serial Interface</i>
FET	–	<i>Field-Effect Transistor</i>
FOV	–	<i>Field Of View</i>
GPIO	–	<i>General Purpose Input Output</i>
GUI	–	<i>Graphical User Interface</i>
I2C	–	<i>Inter-Integrated Circuit</i>
IDE	–	<i>Integrated Development Environment</i>
IGBT	–	<i>Insulated Gate Bipolar Transistor</i>
IMAP	–	<i>Internet Message Access Protocol</i>
IPP	–	<i>Integrated Performance Primitives</i>
ISO	–	<i>International Standardization Organization</i>
IV	–	<i>Infravermelho</i>

JPEG	– <i>Joint Photographic Experts Group</i>
LED	– <i>Light Emitting Diode</i>
MCS	– <i>Manufacturer Command Set</i>
MIPI	– <i>Mobile Industry Processor Interface</i>
MLL	– <i>Machine Learning Library</i>
OpenCV	– <i>Open Source Computer Vision Library</i>
PCB	– <i>Printed Circuit Board</i>
PNG	– <i>Portable Network Graphics</i>
POP3	– <i>Post Office Protocol 3</i>
PWM	– <i>Pulse Width Modulation</i>
RPM	– <i>Rotações Por Minuto</i>
RVM	– <i>Reverse Vending Machine</i>
SMD	– <i>Surface Mount Device</i>
SMTP	– <i>Simple Mail Transfer Protocol</i>
TIFF	– <i>Tagged Image File Format</i>
USB	– <i>Universal Serial Bus</i>

# 1. INTRODUÇÃO

O presente relatório surge no âmbito da Unidade Curricular Tese Dissertação, inserida no plano de estudos do 2º ano do Mestrado em Engenharia Eletrotécnica e de Computadores (MEEC) do Departamento de Engenharia Eletrotécnica (DEE) do Instituto Superior de Engenharia do Porto, sob a orientação Científica do Engenheiro Lino Manuel Baptista Figueiredo.

O relatório evidencia a importância de equipamentos que incitem a responsabilidade social no que concerne à proteção ambiental através da utilização sustentável dos recursos naturais. Considerando a sociedade atual fortemente industrializada, na qual a utilização do plástico adquire números crescentes, surge a necessidade de se lidar com estes resíduos, neste caso particular, com garrafas de plástico. Neste âmbito, surgem as *Reverse Vending Machine* que permitem a recolha de garrafas de plástico possibilitando ao consumidor o retorno monetário do valor atribuído a cada garrafa. Esta medida assume-se como uma das mais utilizadas em países como Alemanha e Inglaterra pelo caráter facilitador e incentivador. Assim, o presente relatório procura incentivar a reciclagem destes produtos através da criação de metodologias que relacionem a reciclagem com a utilização das novas tecnologias. Assim, e visando a clarificação dos leitores no que respeita às *Reverse Vending Machine* serão encontrados nos capítulos subsequentes a importância, funcionalidade, tecnologia utilizada e aplicação no contexto real.

## **1.1. JUSTIFICAÇÃO E APRESENTAÇÃO DO TEMA**

O contexto mundial atual põe em vogue o crescente volume de utilização de embalagens de plástico evidenciando os componentes sanitários, mercadológicos, socioeconómicos e ambientais associados destacando, desta forma, o incremento da reciclagem e tratamento dos materiais. Este considera ainda as implicações ambientais inerentes ao seu descarte não racional pós consumo, assumindo-se como o desafio da reciclagem racional que pressupõe uma abordagem integrada dos processos de transformação das matérias-primas.

Compreendendo a importância da gestão ambiental nas empresas no que concerne à sustentabilidade e promoção de qualidade de vida, a criação de políticas em foco na condição humana, ecológica e ambiental assume-se essencial na exploração de recursos naturais. Conducente a esta realidade é diretamente vinculada a responsabilidade social empresarial, visando o sentido de obrigação para com a sociedade e, desta forma, a proteção ambiental.

## **1.2. OBJETIVOS**

Objetivando a melhoria da gestão ambiental e evidenciando o papel da tecnologia e da inovação tecnológica no quotidiano, definiu-se como objetivo principal a construção de um protótipo capaz de proceder à análise e aceitação de garrafas de plástico. Desta forma, e de encontro com o objetivo anteriormente traçado são definidos os seguintes objetivos gerais:

- Estudar as soluções existentes no mercado;
- Definir os requisitos necessários para o desenvolvimento do protótipo;
- Identificar o hardware necessário ao desenvolvimento de uma solução para o protótipo;
- Estudar a Análise de Imagem;
- Criar um método de identificação de imagem eficaz;
- Desenvolver uma interface gráfica para interação com o utilizador;
- Construir uma máquina capaz de proceder à análise e recolha de garrafas que possuam os critérios exigidos/planeados;
- Implementar a solução proposta e validar em cenário experimental;

### **1.3. ORGANIZAÇÃO DO RELATÓRIO**

Estruturalmente, o projeto encontra-se organizado em quatro capítulos. O primeiro reporta-se à Introdução na qual se encontra o enquadramento espaço-temporal, é realizada a alusão à temática, bem como é explanado, de forma generalizada, o estado ambiental do planeta. No capítulo subsequente, intitulado Estado da Arte, são apresentadas técnicas e tecnologias úteis para o desenvolvimento do presente projeto. Designado como Projeto, o terceiro capítulo aborda a arquitetura do sistema e a descrição de hardware e software. O quarto capítulo, Implementação, explana a implementação e execução do projeto. Intitulado de Resultados, o quinto capítulo retrata os resultados evidenciados dos estudos e cálculos executados bem como dos resultados obtidos com a implementação do protótipo proposto. O último capítulo, Conclusão, apresenta uma breve reflexão sobre a experiência desenvolvida e as dificuldades encontradas, sendo apresentadas recomendações e sugestões que potenciem a melhoria de futuros projetos. As referências bibliográficas encontram-se posteriormente aos capítulos delineados permitindo, de forma rápida e eficaz, a consulta de documentos científicos que serviram de base ao desenvolvimento deste documento. Serão, por último, adicionados alguns Anexos alusivos aos esquemas elétricos e código.

Esta página foi intencionalmente deixada em branco

## 2. ESTADO DA ARTE

Com o intuito de incentivar a reciclagem de forma a minimizar a poluição a nível global foram desenvolvidas as máquinas *Reverse Vending Machine*. Neste capítulo serão demonstradas diversas técnicas utilizadas com vista a contribuir para uma maior eficácia e fiabilidade na recolha e análise da garrafa bem como uma introdução às tecnologias utilizadas neste projeto.

### 2.1. REVERSE VENDING MACHINE – RVM

As *Reverse Vending Machine* (RVM) são equipamentos desenvolvidos cuja funcionalidade se prende com a análise e recolha de garrafas e a consequente devolução do valor monetário correspondente ao valor da garrafa. A utilização deste equipamento representado na Figura 1 centra-se em países cuja legislação ambiental se assume como prioritária, nomeadamente Noruega, Alemanha e Estados Unidos da América. Assim, e no contexto atual, surgem grandes empresas que visam a manipulação adequada dos recursos como a TOMRA, líder Mundial segundo o site oficial da mesma. Existem todavia outras empresas como a Envipco e a Reserve Vending, empresas de referência em diferentes países.

Estes equipamentos apresentam, na sua generalidade, igual princípio de funcionamento: a garrafa é inserida na máquina pelo utilizador e esta, com a ajuda de tapetes rolantes, procede à sua recolha, rodando-a para assim expor o código de barras e/ou símbolo aos sensores



existentes que, por sua vez, conduzem à identificação da mesma. No caso de validação da garrafa – que obedece a diversificados parâmetros, nomeadamente o fabricante, esta é conduzida para um compartimento diferenciado e posteriormente compactada com vista a economizar o espaço existente para o seu armazenamento. Para finalizar o processo descrito, é emitido um talão que apresenta o valor da garrafa(s) introduzida(s) e devolvidas. Na eventualidade da garrafa não corresponder às características delineadas, a mesma é devolvida ao utilizador.

Atualmente o mercado disponibiliza *RVM* tecnologicamente mais evoluídas capazes de proceder à recolha de imagem em tempo real através da comparação com uma base de dados e, desta forma, proceder à decisão de destruição ou reaproveitamento.



Figura 1 – Reverse Vending Machine

## 2.2. ATUALIDADE: TECNOLOGIAS IMPLEMENTADAS

O avanço tecnológico evidenciado, desde a invenção do transístor à conceção de componentes eletrónicos do tipo *Surface Mount Mevice (SMD)*, permitiu dar início ao desenvolvimento de máquinas de menores dimensões, de maior eficiência e consequentemente obtenção de melhores resultados. Assim, a tecnologia aplicada nas *RVM* não sendo única, recorre a diversificadas metodologias de análise conducentes a maior rapidez e diminuição do erro de análise recorrendo para tal, à utilização de inúmeras câmaras estrategicamente aplicadas.

Face ao avanço anteriormente mencionado, diferentes fabricantes detêm e aplicam diversificados sistemas de deteção e análise de recipientes. A TOMRA ([9]), a título de exemplo, utiliza o (1) *Flow Technology*, (2) *Sure Return* e (3) *True Vision*.

O *Flow Technology* procede à análise através da utilização de câmaras fotográficas posicionadas em redor do orifício de entrada, tal como é possível verificar na Figura 2. Quando se procede à inserção da garrafa na máquina, a mesma é colocada sobre duas correias motrizes que, paralelamente, desencadeiam a ativação das câmaras – por norma seis - existentes numa disposição anelar que aquando da passagem da garrafa captam diversos fotogramas e procedem à comparação através da base de dados inserida pelo fabricante. Na eventualidade do detentor da máquina pretender proceder à recolha de garrafas de um determinado fabricante e necessite que a mesma possua o seu rótulo original, este sistema torna-se a melhor solução apresentada visto que a comparação dos fotogramas recolhidos com o existente na base de dados, permitirão decidir se a garrafa será aceite ou rejeitada. Na eventualidade de coincidência da imagem obtida, a garrafa é deslocada pelas correias e conduzida até ao local de armazenagem.

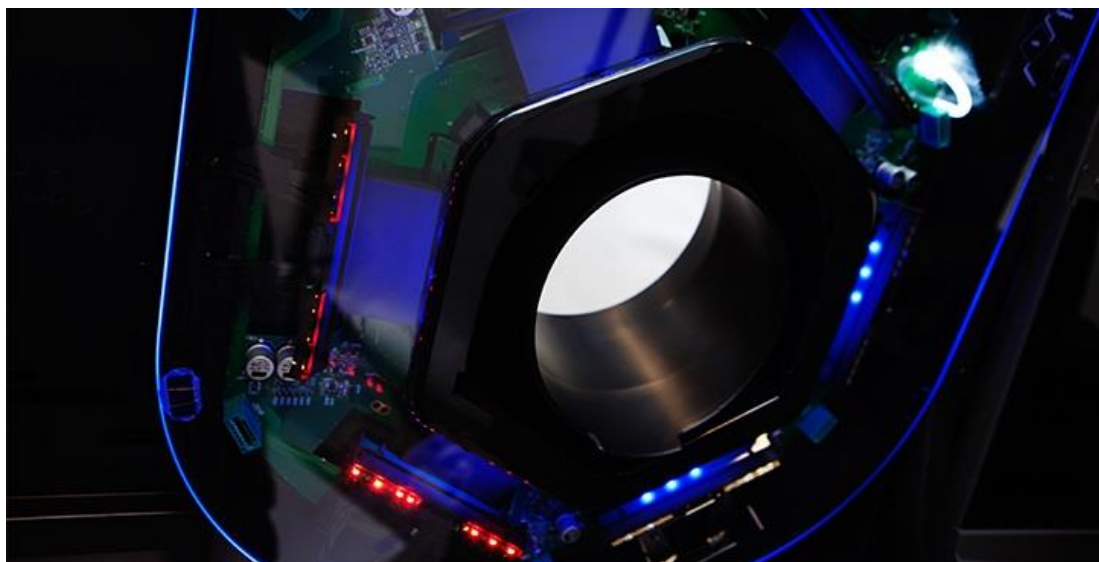


Figura 2 – Flow Technology

O segundo sistema de análise, o *Sure Return*, é um sistema composto por um emissor de luz e uma câmara posicionados quase que sobrepostos e cuja luz, quando emitida, atravessa uma lente de Fresnel com o intuito de direcionar os raios de luz para o objeto a analisar. Numa posição oposta encontra-se um espelho que possibilita o retorno dos raios luminosos para a câmara e consequentemente a emersão do contorno da garrafa, tal como é visível na Figura 3. A partir da obtenção do desenho e da comparação com o existente na base de dados é possível determinar se a garrafa introduzida será aceite ou rejeitada. A utilização desta tecnologia permite verificar, por exemplo, se a garrafa contém líquido no seu interior, visto

que a presença deste ter a capacidade de alterar o ângulo de reflexão da luz e desta forma alterar o desenho da garrafa.

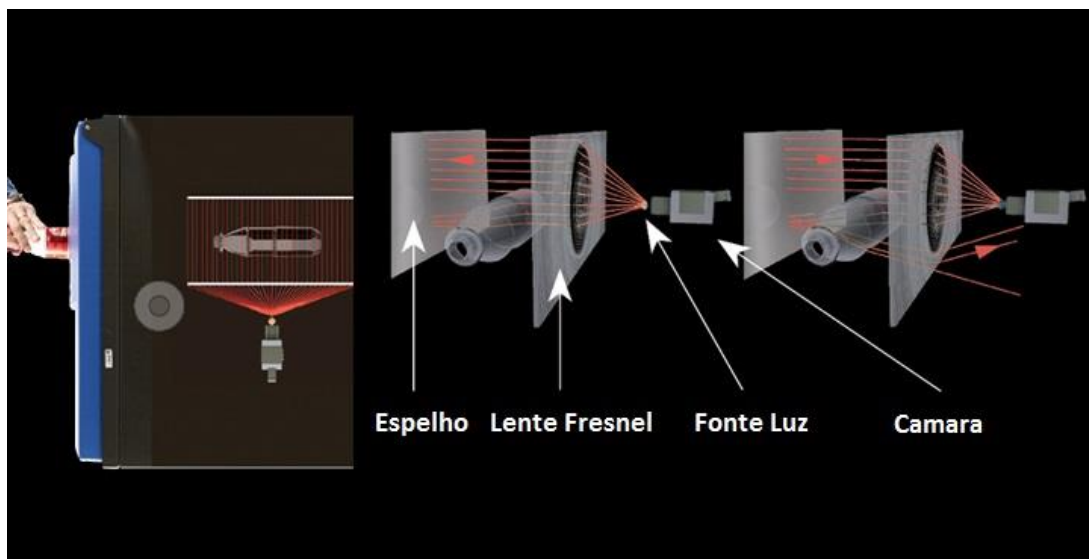


Figura 3 – Sure Return

Ilustrado na Figura 4, o terceiro sistema de análise introduz a capacidade de aceitação de grades de garrafas. Através da recorrência a sistemas óticos emerge a possibilidade de se proceder à contagem do número de garrafas presentes em cada uma das grades. Ostentando como base os sistemas anteriormente descritos, depreende-se que através da recorrência a feixes de luz, a detecção de garrafas se afirma como uma realidade.

Visando a utilização de câmaras de vídeo, com o intuito de proceder à detecção do símbolo demarcado na grade a analisar, é possibilitada a determinação do seu fabricante.

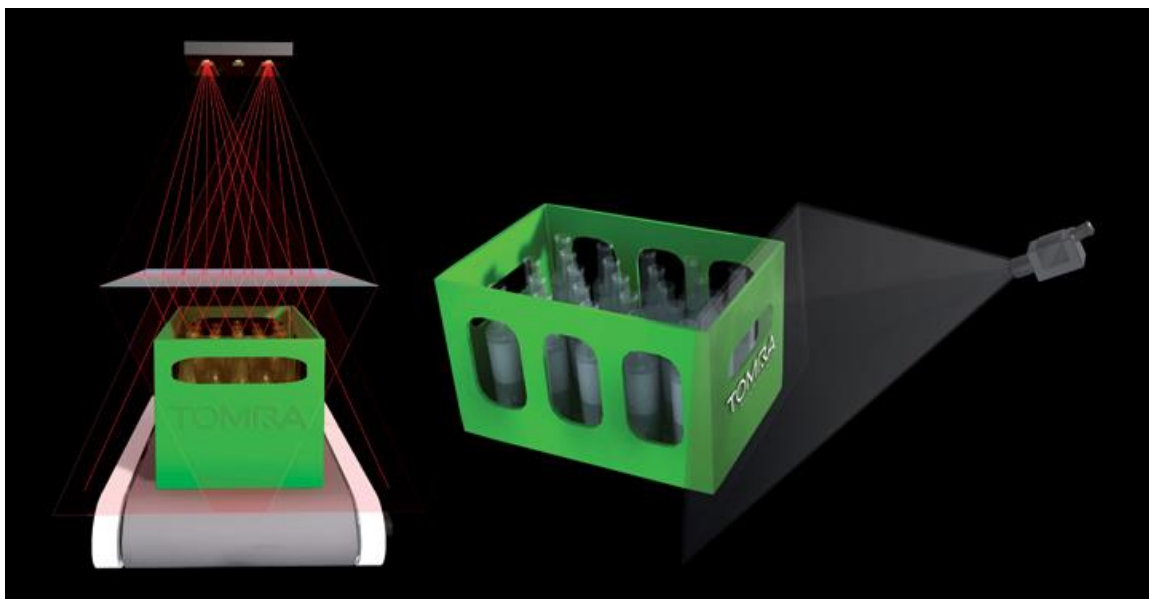


Figura 4 – True Vision

Realizado o processo de análise de imagem (independentemente do selecionado) a garrafa persegue o percurso previsto: ou é aceite ou é rejeitada pela máquina. No caso da mesma reunir as características que permita a sua aceitação, processa-se o transporte realizado por correias (anteriormente mencionadas) que a conduzem ao local de armazenamento. A tipologia de armazenamento pode ser diversificada dependendo do objetivo estabelecido: as garrafas recolhidas podem ser (1) compactadas visando uma menor ocupação espacial; (2) mantidas com a forma original visando, por exemplo, o seu reenchimento ou (3) separadas por formato e/ou fabricante. Face à diversidade de possibilidades, as RMV podem conter o seu próprio compartimento de armazenagem apresentando, na sua maioria, da tipologia 1 anteriormente mencionada. Estas podem ainda apresentar um ou mais locais de armazenagem que podem ser etiquetados e assim agrupar as garrafas aceites por tipologia recorrendo, para isso, a um transportador – como tapetes rolantes – que através da receção de informação do tipo de garrafa inserida, a conduz até ao local próprio de armazenagem. Para que o processo seja realizado com sucesso é necessária a utilização de sensores como as fotocélulas ou interruptores de fim de curso.

## **2.3. TECNOLOGIAS IMPLEMENTADAS NA REALIZAÇÃO E DESENVOLVIMENTO DO PROJETO**

A diversidade de tecnologias no mercado atual conduziu a uma extensa revisão da literatura que, desta forma, possibilitou a uma seleção mais adequada das tecnologias a empregar, mais concretamente no que concerne ao controlo e análise de sistemas. Neste âmbito, o presente capítulo explana tecnologias como sensores de imagem, Raspberry Pi, linguagens e bibliotecas de programação e algoritmo de deteção de símbolos em vídeos.

### **2.3.1. SENSORES**

Designados como (...) *qualquer componente ou circuito eletrónico que permita a análise de uma determinada condição do ambiente (...) como temperatura ou luminosidade* ([11]) os sensores podem ser classificados como um tipo de transdutor – componente que transforma um tipo de energia num outro. O seu desenvolvimento e aplicação trouxe inúmeras vantagens e comodidades à sociedade atual.

#### **2.3.1.1. SENSOR DE IMAGEM**

Os Sensores de imagem apresentam igual princípio de funcionamento do olho. Estes sensores captam a luminosidade das imagens projetadas sobre o sensor iniciando, consecutivamente, a captura de imagens. Estes apresentam na sua constituição milhões de transdutores fotossensíveis capazes de converter a energia luminosa em energia elétrica que, posteriormente, é lida e gravada de forma digital e cuja combinação da informação obtida permite a formação da imagem.

Atualmente existem dois tipos de sensores de imagens: o *Complementary Metal-Oxide Semiconductor (CMOS)*, presente na Figura 5, e *Charge-Coupled Device (CCD)*. O sensor *Complementary Metal-Oxide Semiconductor* é caracterizado pelas suas características físicas que denotam uma construção mais compacta e, consequentemente menor área de utilização para a formação de imagem. Todavia, o sensor CMOS ([4]) permite ao utilizador uma captação de imagens de maior qualidade sendo atualmente utilizados, a título de exemplo, em máquinas fotográficas. Por sua vez, o *Charge-Coupled Device* requer uma maior área física para o processo de formação de imagem apresentando, no entanto, imagens de menor qualidade, mas de obtenção mais rápida quando comparados com os sensores CMOS - fator importante em determinadas tecnologias, como em câmaras de vídeo vigilância.

Visando a obtenção de imagem a cores são utilizados sensores monocromáticos que requerem a aplicação de filtros coloridos como o *Filtro Bayer* explanado na Figura 6, ([1]). Este permite a passagem de luz vermelha, verde e azul em diversificadas zonas e, desta forma, possibilita a incisão da combinação de cores no sensor. A informação obtida a partir deste procedimento é, numa fase posterior, processada.

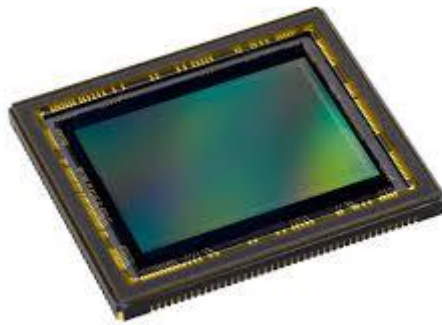


Figura 5 – Sensor CMOS

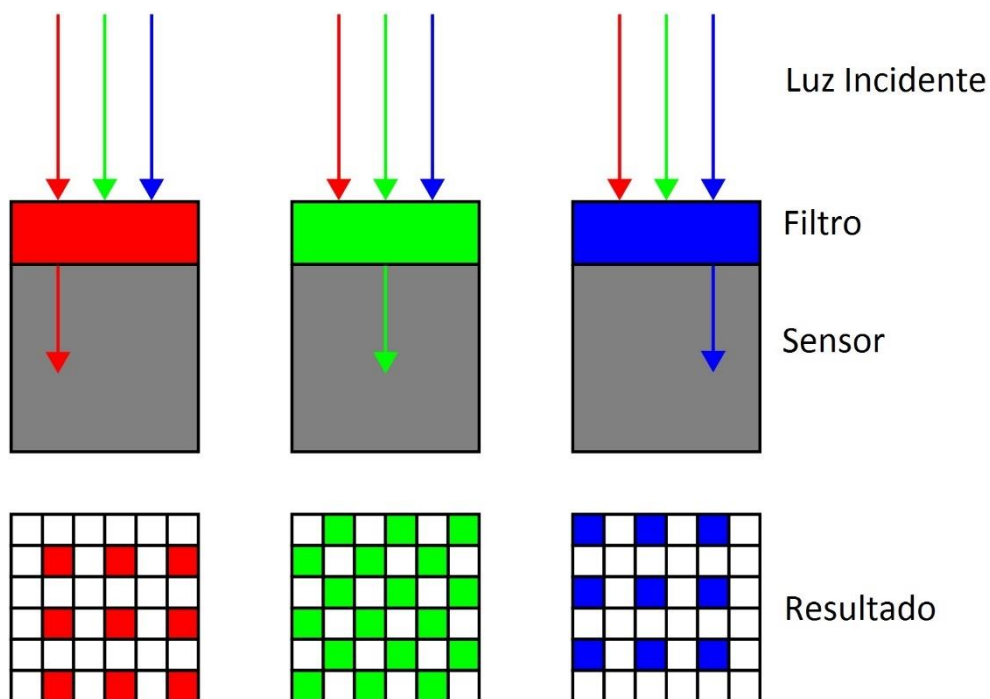


Figura 6 – Filtro Bayer

## 2.4. RASPBERRY PI

*Raspberry Pi* é um microcomputador de baixo custo e com dimensões equivalentes a um cartão de crédito que possui uma unidade de processamento central e partes de condicionamento de sinal – Figura 7. Estes tipos de microcomputadores permitem a instalação de um sistema operativo, maioritariamente o sistema Linux (de distribuição gratuita) bem como a ligação a um ecrã. Estes apresentam ainda uma saída de Áudio, diversas portas USB e uma porta para comunicações do tipo Ethernet bem como pinos de entradas e saídas digitais I/O.

As potencialidades anteriormente mencionadas tiveram como objetivo a promoção do ensino da ciência da computação permitindo aos seus utilizadores criar o que a imaginação permite.

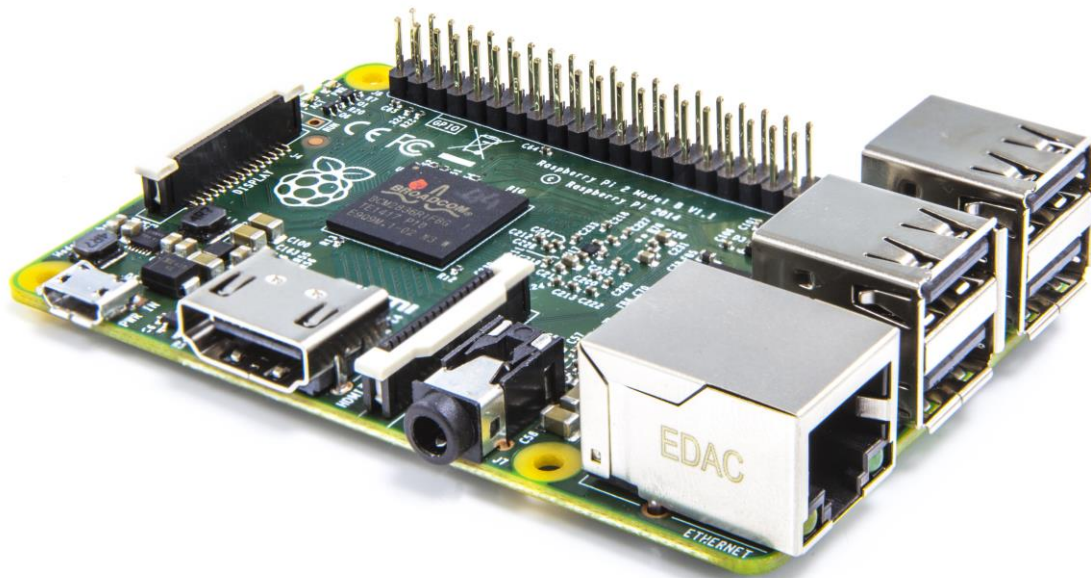


Figura 7 – Raspberry Pi 2

Desde o início da sua comercialização a Raspberry apresentou um elevado sucesso que se refletiu nas milhares de encomendas realizadas. Numa fase inicial foram lançadas duas versões – A e B, cuja diferença centrou-se na Interface de rede por Ethernet. Os primeiros modelos eram caracterizados pela presença de um microprocessador, *Broadcom SoC – ARM1176JZF-S*, de 700MHz que devido à baixa temperatura de funcionamento não requer uma refrigeração externa, sendo desde logo um dos pontos fortes para os seus utilizadores, bem como a sua memória RAM de 128MB alocados para a Graphics Processing Unit (GPU) bem como 128 MB disponíveis para o sistema, ([14]).



## **2.5. LINGUAGENS E BIBLIOTECAS DE PROGRAMAÇÃO**

Considerando o universo dos microcontroladores, o seguinte subcapítulo explana as diferentes e mais relevantes linguagens e bibliotecas de programação, considerando o desenvolvimento do projeto proposto.

### **2.5.1. LINGUAGENS DE PROGRAMAÇÃO**

Com o aparecimento dos computadores (microprocessadores/microcontroladores) as linguagens de programação tornaram-se uma necessidade crescente uma vez possibilitarem a resolução de algoritmos matemáticos de maior complexidade de forma mais rápida comparativamente a uma resolução manual. Neste âmbito, emerge a Programação Assembly cujo código é diretamente entendido pelo hardware. Sendo denominada como uma linguagem de baixo nível que reconhece os valores lógicos 1 ou 0, a mesma permite ao programador a escrita de comandos que substitui os valores de 1 e 0 por uma linguagem compreendida pelo homem. Não requerendo qualquer tipo de conversão em background, esta linguagem caracteriza-se pela sua velocidade de execução. Todavia, esta linguagem não é multiplataforma o que se traduz na necessidade de proceder a configurações específicas para cada plataforma.

Face às limitações da linguagem de Programação anteriormente referida surge a linguagem de programação em C. Criada em 1972 por Dennis Ritchie com a finalidade de permitir a escrita do sistema operativo Unix recorrendo a uma linguagem de relativo alto nível, esta expandiu-se de forma considerável, não só pela sua eficiência, mas pela integração e utilização por Universidades e programadores. A sua grande utilização pelos diferentes organismos/programadores e o surgimento de diferentes versões da linguagem C conduziu à criação de um modelo *standard* – regido pela ANSI- que delineou as especificações da linguagem. ([5])

A evolução da tecnologia conduziu ao surgimento de novas linguagens de programação como C++ (uma variante da linguagem C), Java, C#, Python entre outras.

Python é uma Linguagem de Programação Interpretada disponível para diversificados sistemas operativos. Esta surgiu de ligações com as linguagens C/C++ permitindo ao programador a construção de uma interface gráfica num curto espaço de tempo através da existência de uma livreria que contribui para a construção de Graphical User Interface - GUI.



Considerada como uma linguagem de *scripting*, não requerendo a compilação do código para a criação de um legível para a máquina, esta linguagem destaca-se de tantas outras, tais como C. Esta pode ainda funcionar como uma ligação entre diferentes programas, utilizando um modelo de chamada e resposta de forma a gerir os inputs e redireciona-los para o/os módulos apropriados. Considerada uma linguagem de alto nível, em *background* desempenha atividades que até então era da responsabilidade do programador, como a gestão de memória. ([10])

### 2.5.2. BIBLIOTECAS DE PROGRAMAÇÃO

Parte fundamental do presente projeto comporta a análise de imagem que preconiza a obtenção de imagem real transversalmente à comparação da mesma com uma outra existente numa base de dados. Inerente ao fato descrito, o presente projeto recorre à utilização de uma biblioteca de visão computacional, designada como *processo de modelagem e replicação da visão humana recorrendo a software e hardware*, a *Open Source Computer Vision Library (OpenCV)*.

Denotado por ([12]), como uma das funções primordiais da Visão Computacional, o processo de reconhecimento requer um conhecimento prévio do objeto através da existência de uma base de conhecimento implementada no código ou aprendida através de um conjunto de amostras do objeto em causa, diretamente relacionadas com o reconhecimento de padrões como textura, forma, dimensão e cor.

Visando o anteriormente explanado, ([12]) preconizam a divisão de técnicas de reconhecimento em dois grupos **(1) Estruturais** e **(2) baseado em técnicas que recorrem à teoria de decisão**. O grupo Estruturais, descreve o padrão como forma simbólica e denomina como método a forma como os padrões se relacionam. Em contrapartida, o grupo baseado em técnicas que recorrem à teoria de decisão descreve os padrões como propriedades quantitativas. Os mesmos autores preconizam que pode surgir da conjugação de ambas as técnicas mencionadas uma outra técnica de reconhecimento que pode objetivar-se como uma mais-valia no progresso tecnológico.

Impulsionada nos anos 50, a visão computacional apresentou linhas distintas: (1) replicação do funcionamento do olho humano, (2) replicação do córtex visual e (3) reprodução do funcionamento do cérebro, sendo a primeira a linha de maior sucesso explanado no surgimento de sensores e processadores de imagem, bem como no melhoramento (precisão

e sensibilidade) de câmaras. Todavia, as descobertas suscitadas consideram as redes de informação cerebral extremamente complexas, procurando abordagens diversificadas como o raciocínio Topdown e o Bottom-up, este último de maior impacto/utilização visto imitar o funcionamento do cérebro. Com este raciocínio o computador pode aplicar transformações diversificadas à imagem e descobrir os seus pontos e diferentes perspetivas. Assim surgem as redes neurais artificiais.

Aludindo ao anteriormente mencionado depreende-se, com facilidade, que a reprodução do funcionamento do cérebro humano seria a linha de maiores ganhos, fundamentalmente pela complexidade inerente aos processos de transformação de dados em forma de decisão intrínseca à espécie humana. Sob a perceção de simplicidade e rapidez, o processo elucidado requer, todavia, um elencado conjunto de processos, desenvolvidos desde a divisão do sinal nas diferentes áreas cerebrais e consequente produção de informação conjugado com a informação visual obtida, a título de exemplo, em terminações nervosas conduzindo, consequentemente, à produção de uma decisão/resposta em concordância com experiências vivenciadas.

A literatura revista reporta ao funcionamento do cérebro humano percecionando a complexidade do processo descrito pela elevada velocidade inerente, bem como pela permissão do processo de estímulo/resposta adequada. Torna-se ainda evidente a sofisticação do olho humano pelo controlo intrínseco de luz incidente na retina e, consequente obtenção de uma melhor imagem.

Paralelamente ao olho humano, e segundo ([2]) o processo de modelagem e replicação da visão humana desempenhado por máquinas assume-se como uma complexa atividade pela receção da informação da câmara ou disco sob o formato de números. Na generalidade dos casos, a máquina utilizada não possui reconhecimento integrado, controlo automático de foco e exposição à luz ou informação de experiências anteriores potenciando, desta forma o papel do programador responsável pelo tratamento e cruzamento de informações e, o consequente gerar de saídas/ações pretendidas.

Atendendo aos conhecimentos evidenciados no campo da visão computacional, a biblioteca OpenCV surge com o intuito de aumentar a eficiência computacional, focando as aplicações de tempo real. Escrita nas linguagens de Programação C e C++ e disponível para diversificados sistemas operativos, a destacar Linux, Windows e Mac OSX, a mesma

objetiva o fornecimento de uma infraestrutura de visão computacional de fácil utilização possibilitando, contudo, a criação de aplicações sofisticadas.

Segundo ([2]) a presente biblioteca possui uma sub-livraria de aprendizagem, designada de *Machine Learning Library* (MLL), utilizada maioritariamente em tarefas de visão computacional e dotada de instruções capazes de serem aplicadas numa parte considerável da aprendizagem de máquinas.

Abrangendo uma variedade díspar de funções - destacando a segurança, a medicina e a robótica, a Biblioteca OpenCV, em contacto com a biblioteca *Integrated Performance Primitives* (IPP), permitirá a utilização de processadores *multicore*.

### 2.5.3. AMBIENTE GRÁFICO

A realidade expressa evidencia a importância do ambiente gráfico da aplicação responsável pela interface *Utilizador-Máquina* pela aptidão da sua utilização quer ao nível doméstico quer ao nível industrial. A atualidade denota um melhoramento significativo destas interfaces detentoras de uma maior diversidade de funcionalidades (como o controlo através de ecrãs táteis e a cores, obtenção de informações referentes à máquina em tempo real e a sua parametrização). A linguagem de programação Python correlacionada com a diversidade de bibliotecas como Kivy, PyQt, PyGUI, libavg e wxPython permite o desenvolvimento multiplataforma (Windows, Mac e Linux) destas interfaces.

Atendendo à biblioteca Kivy, direcionada para o desenvolvimento de aplicações com requisitos multi-toque, a título de exemplo aplicações móveis, denota-se que a mesma apresenta como base OpenGL ES 2. Paralelamente às plataformas anteriormente mencionadas, esta biblioteca possibilita ainda o desenvolvimento de aplicações para a plataforma Android e Ios, possibilitando o desenvolvimento de vídeo jogos capazes de suportar animações e interações. Esta biblioteca apresentando ainda suporte para Raspberry Pi.

**PyQt** é uma das ligações Python mais populares do framework Qt. Atualmente existem duas versões PyQt4 e PyQt5, ambas podem utilizar tanto Python 2 como Python 3. Contendo mais de 620 classes abrangendo interfaces gráficas, comunicações em rede, base de dados SQL, Web browsing e muitas outras disponíveis também no Qt. A última versão do PyQt é a versão 5.4 que veio adicionar suporte às seguintes funções QtBluetooth, QtPositioning,

QtMacExtras, entre outras. Recorrendo ao seu IDE próprio, denominado QtCreator, torna-se possível desenvolver todo o software desde programação em C++ e Python até ao desenho do ambiente gráfico ficando tudo mais integrado neste IDE, contudo é possível desenhar apenas o ambiente gráfico e exportar o ficheiro .ui contendo apenas o desenho de cada página. Tornando-se possível importar este ficheiro para qualquer outro software desenvolvido fora do QtCreator, tornando desta forma o desenvolvimento do ambiente gráfico mais simples e mais rápido.

## 2.6. ANÁLISE DE IMAGEM

Atualmente evidenciam-se duas bibliotecas de análise de imagem comumente utilizadas devido à robustez de algoritmo e eficácia: a *TensorFlow* e *OpenCV*. Baseada em redes neurais e tendo como foco da sua utilização a detecção de caracteres em imagens ou vídeos, a *TensorFlow* requer elevada capacidade de processamento visando a resolução de complexas matrizes tridimensionais. Por sua vez, a biblioteca *OpenCV* desenvolvida especificamente com foco na visão computacional, recorre a técnicas mais simples e específicas, permitindo a análise de imagem com recorrência a menores recursos comparativamente à primeira apresentada.

Evocando a importância do processo de reconhecer, ([16]) põe em vogue o reconhecimento por meio de sistemas computacionais de forma automatizada. Impulsionado pela detecção de faces, a primeira etapa do processo de reconhecimento facial prevê a detecção automática de características de uma determinada imagem ou frame de vídeo, muitas vezes influenciadas por fatores como ruídos, variação de iluminação, expressões faciais, imagem de fundo, orientação da cabeça, obstrução ou sobreposição de faces. A metodologia aplicada apresenta, contudo, elevada capacidade no que concerne à precisão, alta taxa de acerto, baixa taxa de falsos positivos e baixo custo computacional. A metodologia evocada assenta em quatro conceitos centrais: (1) Características de Haar e introdução do conceito imagem integral, tais características são responsáveis pela detecção do símbolo ou objeto pretendido, (2) método de aprendizagem de máquina *AdaBoost*, utilizado com o intuito de selecionar a melhor característica e (3) Classificador em cascata para combinação das características selecionadas por *AdaBoost* ao longo de diversos estágios.

### 2.6.1. CARACTERÍSTICAS DE HAAR E IMAGEM INTEGRAL

Amplamente empregue na detecção de objetos e, atendendo às características anteriormente mencionadas, o método de ([16]) assume-se fundamental no desenvolvimento do projeto proposto, uma vez viabilizar a preparação/treino do detetor numa vasta diversidade de objetos. Denominadas de Harr, as *features* utilizadas na framework de detecção de Viola e Jones assumem-se como um conjunto de características retangulares de duas dimensões (2D). Na detecção de objetos a diversidade de características de Haar é cingida a três características: com 2, 3 ou 4 retângulos, exemplificada na Figura 8.

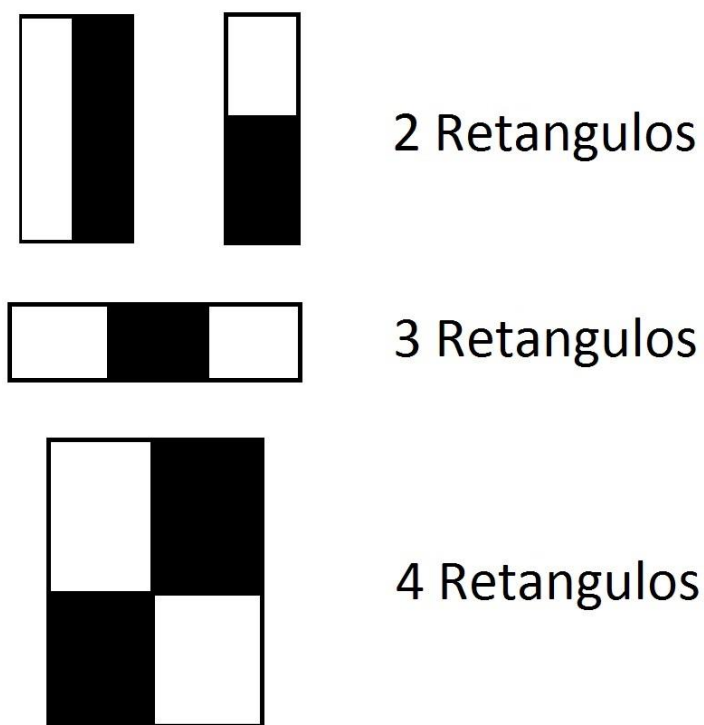


Figura 8 – Características Haar Retângulos

Quando aplicada na imagem, o valor da característica é obtido através do somatório dos pixels contidos na região branca pela diferença do somatório dos pixels contidos na região preta. Visando maior celebridade do processo de análise de imagem aplica-se o conceito de imagem integral em detrimento da utilização da imagem original, representado na Figura 9. Atendendo a que o valor da imagem integral  $(x,y)$ , em que  $x$  representa a coluna e  $y$  a linha, é facultado pela soma dos pixels acima e à esquerda de  $x,y$ , de maneira inclusiva, o valor da mesma é possível obter através da equação 1:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

Onde  $ii(x, y)$  corresponde à imagem integral, Figura 9.

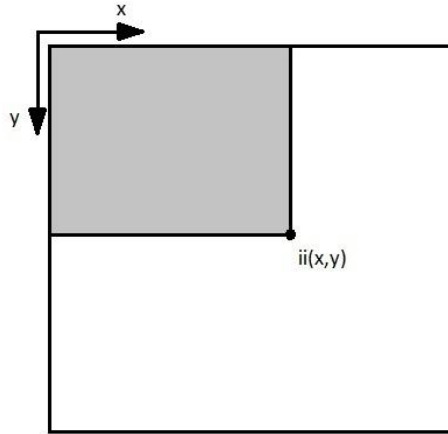


Figura 9 – Imagem Integral

Imputável o baixo custo computacional no que concerne à utilização da imagem computacional, paralelamente à baixa necessidade de acessos à memória e com recorrência a poucas operações aritméticas de soma e subtração, a sua utilização aumenta circunstancialmente. Objetivando a clarificação da mesma, atenda à Figura 10.

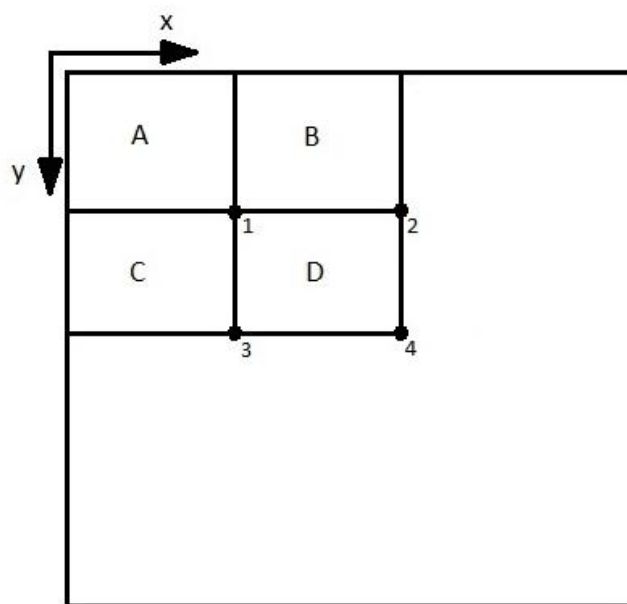


Figura 10 – Cálculo Imagem Integral (imagem adaptada de [16]).

Objetivando o cálculo dos pixels da área do retângulo D, e atendendo que o valor da imagem integral no ponto 1 é obtido pela soma da totalidade dos pixels presentes em A; que o valor de 2 surge de A+B, D é obtido através de A+B+C+D, em suma pela operação  $4+1-(2+3)$ .

A título de exemplo, na presença de uma imagem de 384x288 a análise da mesma recorre a uma digitalização em 12 escalas, em que cada uma apresenta um aumento de 25% em relação à anterior. Viola e Jones, 2004 reportam que cada uma das imagens obtidas passa por um processo de análise que recorre a um detetor de 24x24 pixels. Aludindo à eficiência computacional do processo descrito, a detecção de símbolos pode ser desenvolvida numa taxa de 15 imagens/s.

### 2.6.2. ALGORITMO ADABOOST

Apontada como uma técnica de abordagem à aprendizagem das máquinas, o algoritmo *Adaptive Boosting* idealiza a criação de uma regra altamente precisa que seleciona as características de Haar e treina um combinado de classificadores fracos e imprecisos, propostos por ([7]). Definido como um algoritmo de aprendizagem o mesmo utiliza uma regra de decisão que objetiva a classificação de um exemplo positivo ou negativo, o classificador fraco distingue-se amplamente do classificador forte, cuja obtenção emerge da combinação linear ponderada dos classificadores fracos, visando assim o aumento do

desempenho de detecção. Designado como Boosting (combinação de fatores fracos cujo produto final originado é um classificador forte), o processo descrito canaliza os resultados para a formação de uma cascata de classificadores - estágios de classificadores fortes. Concordantemente com o descrito, o processo de seleção de classificadores (dotada de características) permite, numa fase posterior, a criação de um classificador mais forte, assente na ideia central do algoritmo.

Considerando o contributo imprescindível do algoritmo AdaBoost na resolução de diversificadas dificuldades presentes nos algoritmos de boosting, e segundo o denotado por ([16]) o presente algoritmo pode ser visto como um ávido processo de seleção de características apresentando um conjunto de dados de treino cujas imagens de treinamento são fornecidas (*inputs*) por  $(x_1, y_1), \dots, (x_m, y_m)$  - em que  $x_i$  pertence ao espaço  $X$  da imagem e  $y_i$  assume o valor -1 ou +1, indicativos de imagem negativa ou positiva, respetivamente. O processo em causa recorre repetidamente a um determinado algoritmo básico de aprendizagem a cada período de tempo/ronda ( $t$ ), onde  $t = 1, \dots, T$ .

Baseado numa das principais vertentes do algoritmo – manutenção da distribuição ou verificação do peso de cada interação ao longo de todo o treino, o peso para o treino  $i$  no período  $t$  é denotado por  $D_t(i)$ . Numa fase inicial, a totalidade dos pesos é definida equitativamente sofrendo, a cada ronda, alterações que objetivam os exemplos incorretamente classificados a aumentarem e, desta forma, forçando o algoritmo com o resultado mais fraco a impor-se nos exemplos de maior grau de dificuldade no treino. Segundo ([7]), o trabalho desenvolvido por este procura obter a hipótese, consequentemente, mais fraca  $h_t : X \rightarrow \{-1, +1\}$  apropriada para a distribuição  $D_t$ , dada pela equação 2:

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} D_t(i) \quad (2)$$

Denota-se que o erro é medido relativamente à distribuição de  $D_t$ , na qual o algoritmo mais fraco suportou o processo de treino. Neste processo, este algoritmo recorre à utilização do peso de  $D_t$  ou, na sua impossibilidade, é desenvolvido um subconjunto de amostras de acordo  $D_t$ . De encontro ao anteriormente descrito, e visando a obtenção da melhor hipótese, ([7]) recorrem ao algoritmo de seguida explanado:

Dados:  $(x_1, y_1), \dots, (x_i, y_i)$  onde  $x_i \in X, y_i \in Y = \{-1, +1\}$



Inicialização:  $D_1(i) = \frac{1}{i}$

Para:  $t = 1, \dots, T$

- Treinar o algoritmo fraco usando a distribuição  $D_t$
- Obter hipóteses fracas  $h_t: X \rightarrow \{-1, +1\}$  com erro, através da equação 3:

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \quad (3)$$

- Escolher  $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$
- Atualizar, equação 4:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{se } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{se } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \end{aligned} \quad (4)$$

Onde  $Z_t$  é o fator normalizador.

A hipótese final é dada pela equação 5:

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right) \quad (5)$$

Os processos anteriormente explanados denotam o processo de distribuição de  $D_t$  patenteando o desempenho da mesma aquando do contacto com exemplos (1) correta ou (2) erradamente classificados por  $h_t$ , expondo a forma como  $D_t$  reage com a diminuição do peso ou com o seu aumento, respetivamente. Depreende-se ainda, que a hipótese final  $H$  é a ponderação das hipóteses mais fracas  $T$ , onde  $\alpha_t$  corresponde ao peso atribuído por  $h_t$ .

#### 2.6.2.1. Caraterísticas de AdaBoost e caraterísticas de Haar: Conjugação

Conjugando as caraterísticas de AdaBoost com Haar emerge a possibilidade de se proceder à obtenção de objetos e/ou símbolos no sistema de análise de imagem. Considerando o trabalho de ([16]) e atendendo à Figura 11, é possível observar uma diferença considerável,

no que concerne à intensidade luminosa, na região dos olhos e na parte superior da maçã do rosto, no sentido descendente, passando de uma intensidade mais escura para mais clara.

Tendo em evidência a segunda característica é determinada a diferença de intensidade luminosa entre olhos e nariz caracterizando, com maior intensidade de luz, a região mais central, no caso específico o nariz.

Evidenciando as características enunciadas facilmente se depreende que a recorrência exclusiva a dois classificadores permite detetar a localização dos olhos, de um modo grosseiro, simultaneamente à diminuição do tempo e número de subjanelas a analisar. Importa ressaltar que a técnica utilizada diminui significativamente o número de instruções necessárias para a análise de cada subjanela. De denotar ainda que a prévia combinação de fracos classificadores resultou na escolha das melhores características por forma a analisar a imagem em questão.

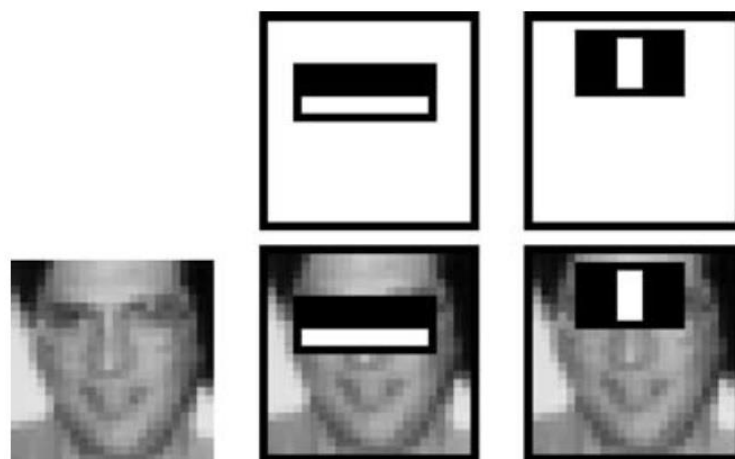


Figura 11 – Características Haar aplicadas a uma imagem [16]

### 2.6.3. CASCATA DE CLASSIFICADORES

Recorrer ao método AdaBoost permite a obtenção de um detetor classificado como bom, com elevada taxa de deteção. Todavia, a utilização de um único detetor é indutivo de baixa velocidade, tornando-se imprescindível o cálculo dos valores das características para toda as subjanelas da imagem.

Visando o incremento do desempenho deste método são concebidos diversos estágios de classificadores eliminando, de imediato e numa primeira fase, parte considerável da imagem ou subjanelas que não correspondem à característica procurada e/ou cujo objeto a detetar

não se encontra presente. Esta etapa verifica-se de importância crucial uma vez contribuir para a diminuição do custo computacional associado e apresentar um aumento de velocidade no processo de deteção. A este conjunto dos estágios de classificadores, denomina-se *Cascata de Classificadores*, exemplificada pela Figura 12.

Considerando cada estágio como um classificador forte e treinado pelo AdaBoost, e designando a entrada inicial como subjanela, na presença de um resultado positivo no primeiro classificador é despoletada a análise do segundo classificador. Todavia, na eventualidade da característica procurada não ser encontrada, a subjanela é rejeitada por um processamento simples. Assim, verifica-se que o tempo computação aumenta em conformidade com o avanço do estágio acompanhando, paralelamente, o aumento da complexidade dos mesmos. À medida que as subjanelas passam pelos estágios denota-se rejeição de uma parte significativa das mesmas, pelo que as capazes de alcançar o último estágio são consideradas deteções.

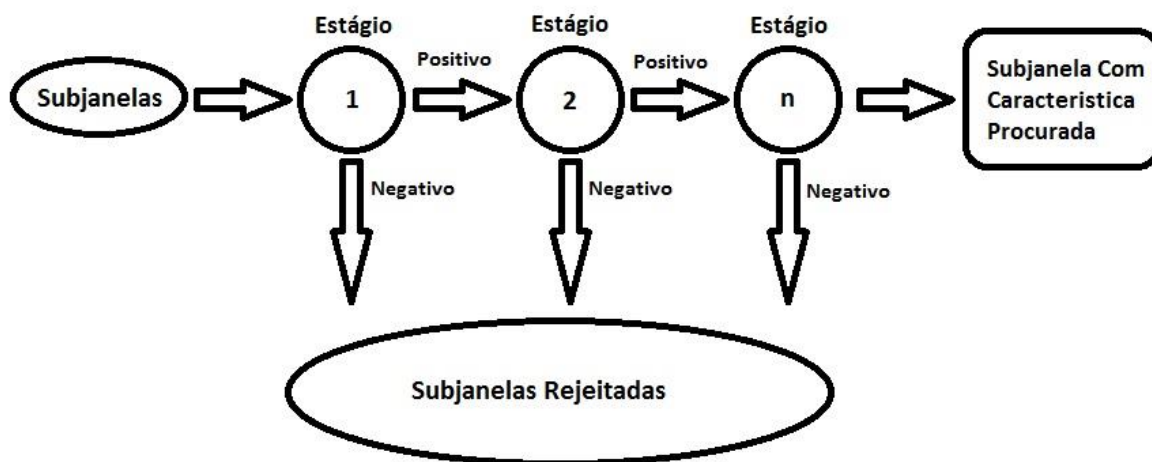


Figura 12 – Cascata de Classificadores

#### 2.6.3.1. Treinar Cascata de Classificadores

Viola e Jones (2004) na formulação do seu algoritmo propuseram uma etapa de treino da cascata de classificadores visando a obtenção de um conjunto adequado de classificadores. Para o procedimento em causa torna-se necessário um conjunto de imagens, intituladas de positivas, que possuam o objeto/símbolo a ser detetado. Considera-se ainda essencial, para cada uma das imagens positivas, a aquisição de coordenadas relativas à posição do símbolo

a detetar. Todavia, e tal como caracterizado pelos autores, é necessária a presença de imagens negativas cujo objeto de interesse não se encontra presente – normalmente imagens de fundo/paisagens. Objetivando a transformação de imagens negativas em positivas, a inserção do símbolo a detetar deve ser introduzida em diferentes ângulos, visando uma deteção mais precisa, Figura 13.

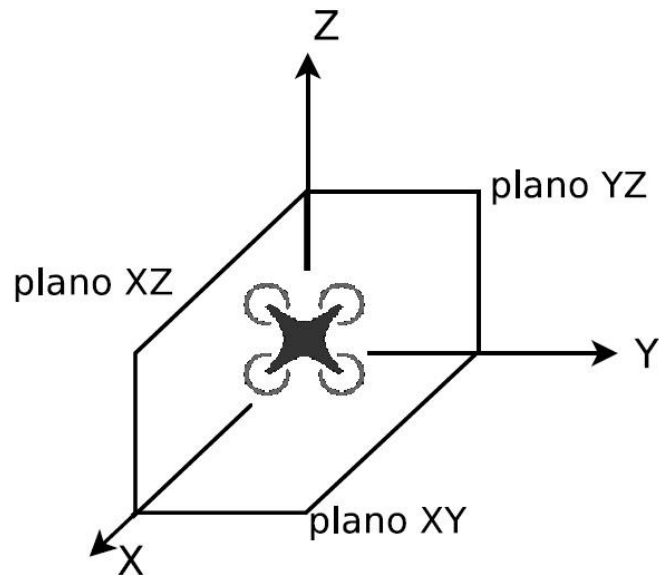


Figura 13 – Plano de rotação do símbolo

Visando a diminuição do tempo de processamento, a totalidade de imagens – independentemente da sua classificação como positivas ou negativas, sofre o processo de conversão para tonalidades de cinza.

Embora denotadas as vantagens da utilização de uma cascata de classificadores, pela sua eficiência e velocidade de deteção, a sua construção denota-se como um processo demorado.



## 3. PROJETO

O presente capítulo aborda a arquitetura geral do sistema através da análise detalhada da tecnologia, atendendo à revisão sistemática da literatura. Serão abordadas temáticas como utilização de sensores infravermelhos, reguladores de tensão, optoacopladores, o controle de motores e a comunicação entre dispositivos, bem como o sistema operativo Raspberry PI.

### 3.1. REQUISITOS

Visando uma melhor compreensão do aspeto geral do protótipo a desenvolver, bem como validando a sua funcionalidade no aspeto mecânico, procedeu-se ao seu desenho numa versão em três dimensões, como explanado na Figura 14.

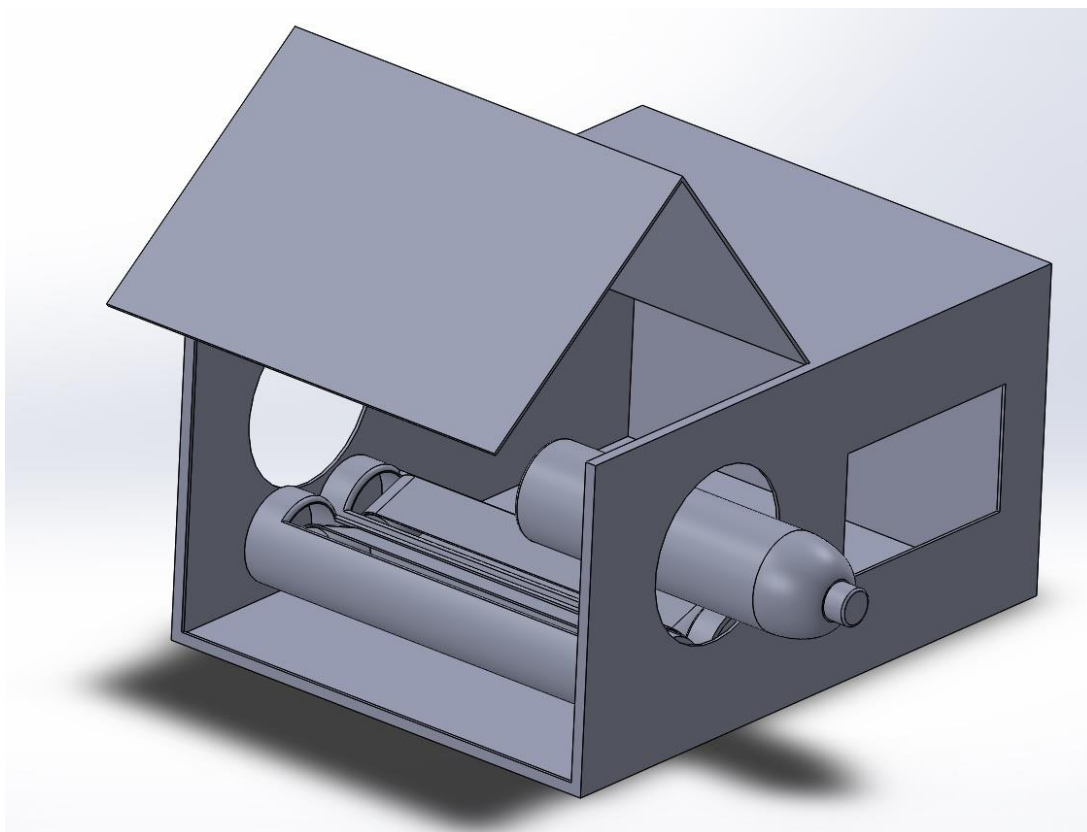


Figura 14 – Modelo 3D Protótipo Mecânico

Objetivando a recolha de garrafas de plástico o projeto a desenvolver visa a implementação de diversificados processos/etapas. Este é iniciado através da implementação de dois tapetes rolantes capazes de procederem à deslocação da garrafa - por sua vez inserida pelo utilizador, e conduzindo-a ao local determinado para proceder a sua análise. O processo de deslocação anteriormente mencionado evidencia a necessidade da implementação de sensores de deteção desde o local destinado à sua inserção na máquina até ao local destinado para a sua análise. Por sua vez, e atendendo ao processo de análise de imagem e à determinação da aceitação ou rejeição da garrafa, proceder-se-á análise do rótulo visando a deteção do símbolo selecionado.

Considerando ainda a câmara responsável pela análise e denotando a inexistência de uma visão de 360 ° pela mesma, evidencia-se a necessidade de se proceder à rotação da garrafa em torno do seu próprio eixo durante o processo mencionado comprovando, desta forma, a necessidade de aplicação de um segundo motor cuja responsabilidade versa no acionamento dos dois rolos que imprimirão, consequentemente, velocidade tangencial à garrafa.

Terminado o processo de análise, e na eventualidade da garrafa ser aceite, a mesma será direcionada para um compartimento de armazenamento. Este, e quando atingida a sua capacidade máxima, procederá ao envio de um email para um endereço eletrónico, a definir, notificando a necessidade manutenção da máquina.

Para a realização deste projeto verificou-se a necessidade de se proceder a diversificados cálculos cuja incidência da sua resolução denota fatos importantes no desenvolvimento do mesmo, a denotar: a distância necessária da câmara ao objeto a analisar e a velocidade de rotação da garrafa para o correto funcionamento do hardware.

### 3.1.1. DISTÂNCIA DA CÂMARA AO OBJETO

Considerando o ângulo de visão da lente existente na Camara de vídeo surge a possibilidade, através da realização de alguns cálculos, de determinar a área efetiva captada pela mesma. Atendendo ainda às características fornecidas pelo fabricante torna-se possível a aquisição do ângulo de visão ou FOV, horizontal e vertical que, considerando a máquina selecionada, apresentam-se de 53.5° e de 41.4°, respetivamente.

Visando a colocação da câmara a uma distância ideal do objeto a analisar, como explanado na Figura 15, em que  $\theta$  designa o ângulo da lente da camara,  $d$  a distância da camara ao objeto,  $u$  o comprimento do objeto detetado pela camara, e atendendo a que o símbolo pretendido se encontra impresso no rótulo, revela-se essencial a determinação das dimensões do mesmo. Confirmada a presença do símbolo a analisar no rótulo das garrafas importa garantir que as mesmas se encontram posicionadas a uma distância mínima garantindo assim a captação do símbolo pela câmara.

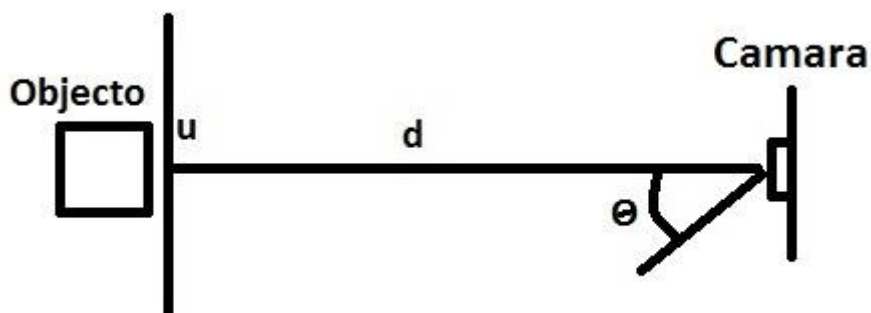


Figura 15 – Distância Camara ao Objeto



Considerando a garrafa de 2l como a maior a sofrer o processo de recolha obtém-se dados como a altura e largura do seu rótulo, 80mm e 320mm, respetivamente. A dedução dos dados apresentados percebe-se que o ângulo de visão vertical restringe a altura mínima, uma vez apresentar uma maior redução e, desta forma, requerendo uma maior distância no processo de análise do rótulo.

A obtenção dos parâmetros anteriormente referidos permitiu a execução dos cálculos referentes a **d** (distância da câmara ao objeto) na presença quer do ângulo vertical quer do ângulo horizontal, como explanado de seguida, respetivamente.

#### **Na presença do ângulo vertical:**

A divisão do ângulo vertical da lente, em metade, conjuntamente com o conhecimento da altura referente ao rótulo da garrafa de maiores dimensões (no presente projeto, 2l) possibilita determinar a distância ideal na qual a câmara se deve localizar objetivando a análise do rótulo na sua totalidade, equação 6.

$$d = \frac{u}{2 * \tan(\theta)} \quad (6)$$

Assim, e através da recorrência a razões trigonométricas deduz-se que para um rótulo de 80mm de altura a câmara a utilizar tem a necessidade de se encontrar a uma **distância mínima de 105mm**.

#### **Na presença do ângulo Horizontal:**

Recorrendo à utilização do ângulo Horizontal, considerando que o rótulo apresenta 320mm de largura e atendendo à distância anteriormente calculada, percebe-se que apenas **105.8mm do rótulo são analisados**, equação 7, tornando necessária a obtenção de um número superior de imagens.

$$\frac{u}{2} = d * \tan(\theta) \quad (7)$$

### **3.1.2. VELOCIDADE DE ROTAÇÃO DA GARRAFA**

Visando a aquisição total da imagem do rótulo pela câmara a utilizar importa a obtenção da velocidade máxima a que a garrafa pode ser rodada. Considerando a maior garrafa alvo de análise, no caso em particular a garrafa de 2l, e atendendo quer ao perímetro total do rótulo

- de 320mm, quer à distância máxima ideal a que a câmara se encontra para a leitura do mesmo - de 105mm, verifica-se uma deteção ineficiente da imagem do rótulo de apenas 105.8 mm da sua totalidade. Verificando a impossibilidade de análise total do rótulo emerge a necessidade de obtenção de diversas imagens do mesmo. Dividindo o perímetro total do rótulo pela capacidade de análise da câmara localizada à distância ideal obtém-se o número total de imagens a capturar,  $\frac{320}{105.8} = 3.02$ . Verificado um resultado superior à unidade, e com o intuito de não perder nenhuma das partes que constituem o rótulo, o valor sofrerá um arredondamento por excesso perfazendo **um total de 4** imagens necessárias.

Assente quer nos parâmetros anteriormente explanados quer dos dados obtidos através do desenvolvimento de um programa, presente no Anexo D, na Linguagem de Programação Python – responsável pelo início de deteção do símbolo e medindo, consequentemente, o tempo dispensado para a execução da imagem e sua análise, tornou-se possível a dedução dos seguintes parâmetros:

$$\text{Número de imagens a obter por rotação} = 4$$

$$\text{Tempo de obturação e análise de uma imagem} = 0,19\text{seg}$$

$$\text{Tempo necessário para analisar uma garrafa} = 4 * 0,19 = 0,76\text{seg}$$

Definida como a *razão entre o deslocamento angular pelo intervalo de tempo do movimento*, a velocidade angular visa a obtenção do objetivo traçado no presente subcapítulo: a **obtenção da velocidade máxima a que a garrafa pode ser rodada**. Alcançada através da Fórmula Matemática, equação 8:

$$\omega = \frac{2\pi}{\tau} \quad (8)$$

Onde **w** designa a velocidade angular e **t** o tempo, obtém-se que a **velocidade de rotação ideal** da garrafa de 2l será de **8.27rad/s** ou **79RPM**.

Atendendo ao projeto em desenvolvimento e considerando que a sua aplicação é baseada na passagem de velocidade dos rolos para a garrafa por forma tangencial surge a necessidade de se proceder ao cálculo da velocidade angular dos rolos considerando as diferenças de diâmetro entre os mesmos e a garrafa a analisar, como exemplificado na Figura 16.

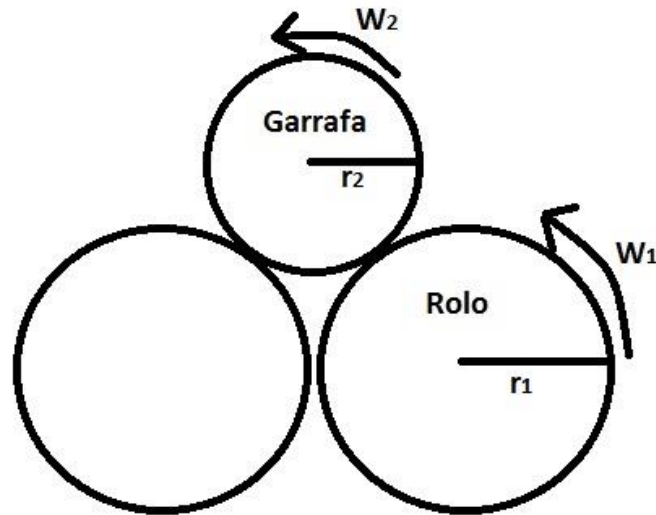


Figura 16 – Cálculo da Velocidade dos Rolos

Tendo em conta que o raio de cada rolo é de  $40mm$  ( $r_1 = 40mm$ ) e da garrafa é de  $50mm$  ( $r_2 = 50mm$ ), sabendo que a velocidade angular da garrafa deverá ser de  $8,27rad/s$  ( $\omega_2 = 8,27rad/s$ ) torna-se possível através da equação 9 proceder-se ao cálculo da velocidade angular dos rolos.

$$\omega_1 \times r_1 = \omega_2 \times r_2 \quad (9)$$

Através da resolução da equação 19 obtém-se que  $\omega_1 = 99$ , logo a imposição de uma velocidade de 99RPM nos rolos impõe à garrafa de 2l a realização de **uma volta completa** num tempo igual a 0,76s, como calculado anteriormente.

Considerada a maior garrafa a sofrer o processo de análise e delineando-a como modelo, denota-se a necessidade de manter a velocidade praticada para a sua análise, como padrão, uma vez que as garrafas com possibilidade de sofrerem o processo de análise poderem apresentar menor raio executando, consequentemente, rotações completas num tempo inferior.

### 3.2. ARQUITETURA GERAL

O Projeto desenvolvido objetiva a recolha de garrafas de plástico recorrendo à implementação de um microprocessador capaz do desempenho de diversificadas tarefas, de forma simultânea e com um grande poder de processamento: *Unidade de processamento central*. Em conotação com esta é possível encontrar-se diversas unidades com funcionalidades variadas, a destacar: **camara**, responsável pela captação de vídeo (numa fase primária uma captação em bruto uma vez ser, numa fase posterior, analisada pela unidade de processamento central), **consola tátil**, que permite a apresentação de informações relativas à máquina bem como à interação entre a interface máquina-utilizador, **sistema de recolha** das garrafas, responsável pelo controlo de motores bem como leitura dos sensores necessários ao correto funcionamento, e envio de **email** que permite a notificação no caso de ser atingida a sua capacidade máxima.

Desta forma, e visando um funcionamento pleno dos diversos módulos são utilizados protocolos de comunicação distintos entre estes e a unidade de processamento central. Salientam-se, todavia, os protocolos *Camera Serial Interface* (CSI) - responsável pela comunicação entre a câmara e a unidade de processamento central, o *Display Serial Interface* (DSI) que estabelece a ligação entre a consola tátil e a unidade de processamento central, o *Simple Mail Transfer Protocol* (SMTP) que procede ao envio de um email sempre que o mesmo seja desejado. Por sua vez, e para se proceder ao controlo e leitura de sensores do sistema de recolha, recorre-se à utilização de pinos de *General Purpose Input Output* (GPIO) presentes na unidade de processamento central.

Visando uma melhor perceção do proposto pelo presente projeto é apresentado um diagrama de blocos, explanado na Figura 17.

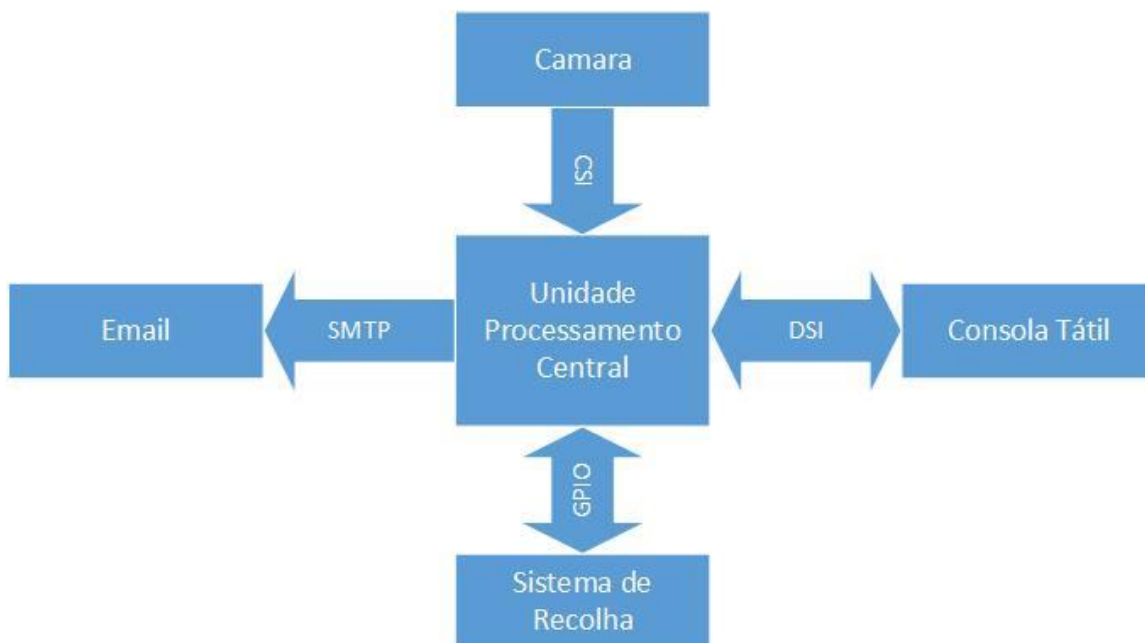


Figura 17 – Arquitetura Geral do Sistema

### 3.2.1. SISTEMA DE RECOLHA

Com o intuito de explicitar o trabalho proposto e de encontro ao anteriormente referido, surge o Bloco do sistema de recolha explanado na Figura 18, cuja responsabilidade depreende a movimentação e controlo das partes mecânicas - mais concretamente o tapete rolante e rolos, bem como a leitura da totalidade dos sensores presentes na máquina (o sensor de entrada e saída da garrafa).

Denota-se que as saídas digitais da unidade de processamento central controlam o circuito integrado responsável pelo acionamento dos respetivos motores conduzindo, conseqüentemente, ao acionamento das partes móveis da máquina. Não obstante, as mesmas permitem o acionamento de transístores que possibilitam a alimentação de dispositivos elétricos, a título de exemplo, luzes que serão estrategicamente posicionadas. Por sua vez, e no que concerne às entradas digitais na unidade de processamento central, torna-se possível a recolha dos sinais emitidos pela totalidade dos sensores, imprescindíveis ao correto funcionamento da máquina.

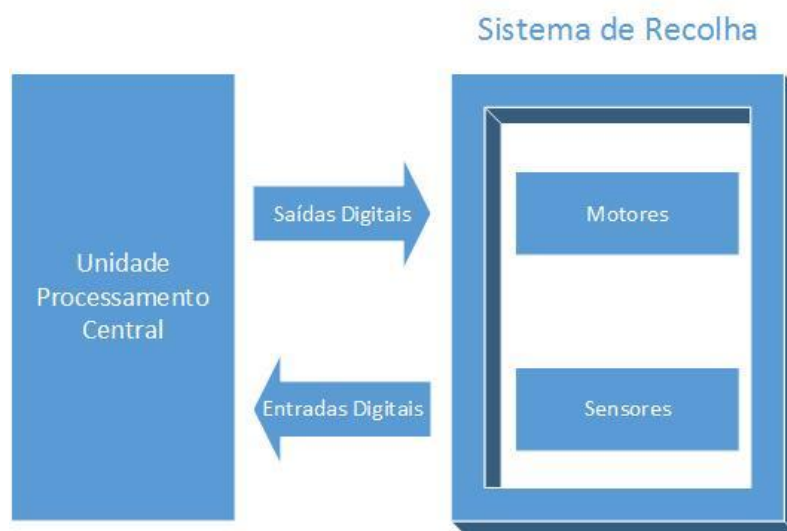


Figura 18 – Arquitetura Geral do Sistema – Sistema de Recolha

### 3.3. DESCRIÇÃO DA TECNOLOGIA UTILIZADA

Considerando, de forma generalizada, o hardware como um conjunto de componentes físicos e projetando futuras intervenções no projeto proposto, quer numa perspetiva de manutenção quer em melhoramentos posteriores, a presente secção do projeto aborda e explicita os diversificados componentes eletrónicos necessários para a construção do protótipo final.

#### 3.3.1. UNIDADE DE PROCESSAMENTO CENTRAL

A unidade de processamento central assume-se uma parte integrante e essencial do projeto, sendo mesmo considerada como o núcleo de todo o protótipo e local destinado para a execução do seu software. Assim, e em conformidade com o descrito, a mesma terá de reunir um conjunto de requisitos específicos, de realçar, a **elevada capacidade de processamento** que objetiva a execução de diversificadas funcionalidades – a título de exemplo a análise em tempo real e gestão de entradas e saídas digitais, a **boa capacidade gráfica** que visa o fornecimento de um ambiente gráfico capaz de interagir com o utilizador. Importa ainda realçar a **diversidade de comunicação** como um outro requisito relevante, uma vez o protótipo a desenvolver requerer a ligação à internet por ou sem cabo, visando o envio de um email quando o mesmo se considerar fundamental. A capacidade de conexão à câmara de vídeo e à consola tátil assume-se também como requisito importante para a unidade de processamento central.

Atendendo aos requisitos enumerados e consequente investigação de diferentes unidades de processamentos, a seleção recaiu no microcomputador Raspberry Pi. A presente escolha emergiu quer das suas características intrínsecas, quer pelo baixo preço, versatilidade e suporte existente.

### 3.3.2. COMUNICAÇÃO ENTRE DISPOSITIVOS

Objetivando o funcionamento uníssono do sistema, importa ressaltar a importância da interligação dos componentes como a câmara de vídeo à unidade de processamento central ou o ecrã à Raspberry pi.

A existência de comunicação entre os módulos é realizada através da existência de protocolos de comunicação, isto é, um conjunto de especificações que caracterizam o formato, a sincronização, sequências bem como a deteção de erro. No que concerne à comunicação **Raspberry Pi – Camara**, e segundo ([3]) são utilizados os protocolos *Inter-Integrated Circuit (I<sup>2</sup>C)* e *Camera Serial Interface (CSI)* cujas finalidades são diversificadas. O primeiro, caracterizado pela baixa largura de banda, visa o transporte dos comandos e da informação relativa às configurações do processador ao sensor de imagem e recorrentemente utilizado em diversificados comandos, a título de exemplo, iniciar ou parar o sensor, alterar a configuração de resolução de saída do sensor bem como o controlo do tempo de exposição do sensor. Por sua vez, o protocolo CSI comporta uma largura de banda superior e, sendo ele também uma comunicação série, é responsável pelo transporte de dados da imagem obtida para o processador localizado na Raspberry. Todavia, a ligação física Raspberry Pi-Câmara de Vídeo é realizada através de uma fita de condutores.

No que respeita à ligação **Raspberry – Consola Tátil**, recorre-se à utilização do protocolo de comunicação *Display Serial Interface (DSI)* conjuntamente com uma interligação física, procedida com a utilização de uma fita de 15 condutores. Visando o fornecimento direto dos dados gráficos para o ecrã, o mesmo ocorre através do conector S2, como explanado na Figura 19.

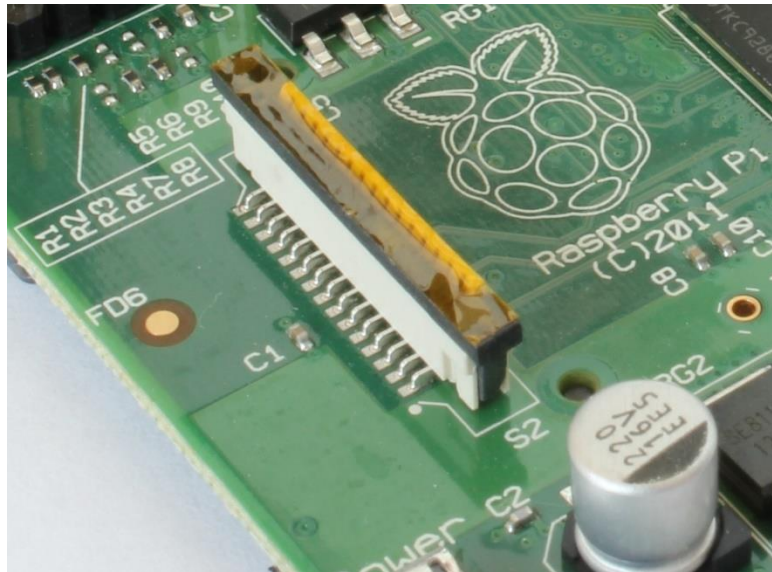


Figura 19 – Conector DSI

Desenvolvido pela *Mobile Industry Processor Interface (MIPI)*, o protocolo DSI surgiu com o intuito de minimizar os custos dos controladores de ecrã para dispositivos móveis ou dispositivos cujo espaço de construção requer-se o mais diminuto possível. Caracterizado pela definição do barramento série e o protocolo de comunicação entre a fonte da imagem e o dispositivo de destino, o protocolo DSI define, ao nível da ligação física, uma ligação série ponto a ponto de elevada velocidade que engloba quer uma faixa *clock* de alta velocidade quer uma ou mais faixas para dados. Estas, por sua vez, apresentam na sua constituição dois fios adaptando-se, desta forma, ao par diferencial de sinais.

Evidenciando a comunicação existente no protocolo anteriormente mencionado, verifica-se que o mesmo pode ser procedido de dois modos distintos: (1) baixa potência e (2) alta velocidade. Considerando o modo (1) **baixa potência**, verifica-se que o *clock* de alta velocidade é desativado e a informação do mesmo torna-se parte integrante nas faixas de dados. Neste, a taxa de dados assume-se como insuficiente para o envio de dados da imagem para o ecrã podendo, contudo, ser utilizada no envio de configurações e/ou comandos. No que concerne ao modo (2) **alta velocidade**, o *clock* – que apresenta o modo de funcionamento de um bit clock para as faixas de dados, é ativado possibilitando o envio de dados referentes à imagem a apresentar no ecrã. Independentemente do modo selecionado atesta-se que a velocidade do *clock* varia diretamente com os requisitos do ecrã a utilizar.

O protocolo de comunicação DSI pode ser subdividido em duas frações: *Display Command Set (DCS)* e *Manufacturer Command Set (MCS)*. A fração DCS do protocolo comporta o



conjunto de comandos que objetivam o controlo do ecrã, permitindo ações como a inversão da imagem no ecrã, a ativação do mesmo ou a seleção do modo *sleep*. Num outro prisma surge a fração MCS que define um conjunto de comandos específicos referentes a cada fabricante incluído, de forma generalizada, comandos responsáveis pela programação de memória não volátil ou capazes de proceder ao controlo da correção de *gamma* da imagem. No que concerne ao formato dos dados enviados pelos métodos mencionados, o mesmo é definido pelo protocolo DSI. Por sua vez, os dados de imagem são desenhados no ecrã em tempo real não sendo procedido o seu armazenamento pelos dispositivos e permitindo, desta forma, a utilização de ecrãs de maior simplicidade sem capacidade de armazenamento de informação. Embora se assuma como vantagem, o dispositivo em utilização tem a necessidade de proceder a atualizações de imagem a pelo menos 30 imagens/s, salvo a perda de imagens. ([13])

Considerando a necessidade de envio de email, e atendendo ao teor simples e/ou informal que o mesmo requer, denota-se que o protocolo *Simple Mail Transfer Protocol* (SMTP), parte integrante da camada de aplicação TCP/IP, se assume como o indicado para a finalidade pretendida, permitindo o envio de texto sem a inclusão de anexos ou imagens.

### **3.3.3.       SENSORES INFRAVERMELHOS**

Representado pela totalidade das ondas ou partículas eletromagnéticas de diferentes frequências, o Espectro eletromagnético apresenta-se dividido em diferentes zonas podendo ser, em alguns comprimentos de onda, captado pelo olho Humano. Considerada como uma radiação não ionizante na porção invisível do espectro, a luz Infravermelha (IV) pode ser percebida como calor. No protótipo em questão estes têm como finalidade a deteção de entrada e saída da garrafa na máquina.

No que concerne ao nível eletrónico, os emissores infravermelhos assumem-se como díodos emissores de Luz (Light Emitting Diode – LED) no espectro dos infravermelhos e com um comprimento de onda de 950nm. Assim, e com o intuito de proceder à sua captação recorre-se aos detetores, como denotado na Figura 20.

Considerando os Detetores como foto transístores, mais especificamente, transístores bipolares com ausência de base - a mesma substituída pela incidência de radiação luminosa, verifica-se que a incidência de luz infravermelha nesta produz um aumento da corrente de base

implicando, diretamente, uma variação na corrente do coletor proporcional à intensidade de luz incidente.

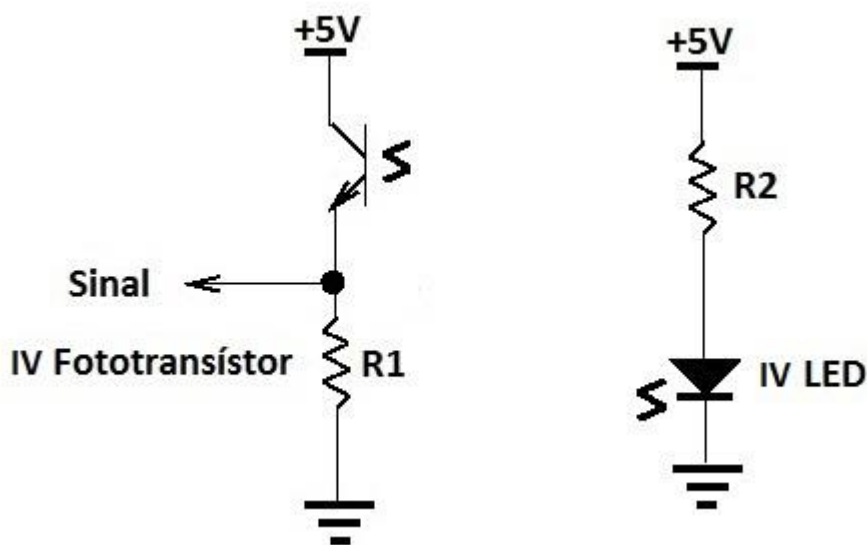


Figura 20 – Detetor e Emissor infravermelho

#### 3.3.4. REGULADORES DE TENSÃO

Os reguladores de tensão são dispositivos cuja finalidade se prende com a manutenção da tensão de saída (DC) de um circuito elétrico subdividindo-se em duas tipologias: Lineares ou Comutados.

Um regulador linear de tensão objetiva o fornecimento de uma tensão de saída estável e constante, independentemente das variações de tensão na entrada, temperatura e carga. Este apresenta uma base constituída por dispositivos ativos, como um transistor de junção bipolar ou de efeito de campo, capazes de funcionar na sua região linear, pelo que a sua regulação é realizada de forma a agir como uma resistência variável capaz de proceder a ajustamentos do divisor de tensão, possibilitando a sua saída constante. Apesar dos reguladores de tensão comutados incluírem também a regulação de tensão, realizada maioritariamente através da modulação da largura de Pulso que controla o corte e a condução do transistor, estes apresentam variações no ciclo de trabalho.

Em termos comparativos, verifica-se que embora os reguladores de tensão linear apresentem um ruído de saída efetivamente inferior aos reguladores comutados com as mesmas características de tensão e corrente de saída, os mesmos apresentam eficiência inferior pela

dissipação de energia em forma de calor, num valor de cerca de 40%, considerando que reguladores lineares apresentam eficiências na ordem de 50% e os comutados na ordem dos 90% bem como correntes de 1A ou 1.5A contra 30A, respetivamente ([15]). Todavia, os reguladores de tensão comutados apresentam alguns fatores de compromisso a serem considerados como preços mais elevados, diminuta oferta pelo mercado bem como a necessidade de criação de circuitos extra de controlo. Assim, para o desenvolvimento do projeto foram utilizados reguladores lineares.

Atualmente, e perante a diversidade de reguladores de tensão existentes no mercado, existe a possibilidade de aquisição de reguladores de tensão lineares sobre a forma de Circuito Integrado (CI) sem necessidade de projeção dos mesmos. Caracterizados pela sua precisão e forma compacta, foi selecionado para o projeto o modelo 78XX para a regulação de tensões positivas. Este regulador apresenta variantes díspares, como explanado na Tabela 1, baseada nos valores fornecidos pela Texas Instruments.

**Tabela 1 Regulador de Tensão**

Positivo	Tensão máxima de entrada (V)	Tensão de Saída Regulada (V)	Corrente máxima de Saída (A)
7805	7 – 25	5	1.5
7808	10.5 – 25	8	1.5
7810	12.5 – 28	10	1.5
7812	14.5 – 30	12	1.5
7815	17.5 – 30	15	1.5
7824	27 – 38	24	1.5

Considerando o regulador selecionado verifica-se que o mesmo não apresenta partes móveis, conduzindo à inexistência de desgaste com o decorrer do tempo. Este apresenta ainda uma tensão de saída fixa, com um erro inferior a 5%. Todavia, para que a mesma se mantenha, o valor de tensão de entrada tem a necessidade de se encontrar entre os parâmetros explanados na Tabela acima exposta.

### 3.3.5. OPTOACOPLOADORES

Considerando a segurança como um ponto-chave no desenvolvimento do presente projeto, a utilização de Optoacopladores assumiu-se como imprescindível, uma vez utilizarem-se diferentes tensões em igual circuito. Responsáveis pelo isolamento elétrico de dois circuitos, os optoacopladores são constituídos por um LED emissor de luz infravermelha e um semiconductor fotossensível (capaz de detetar a luz infravermelha emitida), confinados num pequeno circuito integrado.

O funcionamento dos optoacopladores, exemplificado na Figura 21, inicia com a aplicação de um díodo emissor de infravermelho no terminal de entrada que emitirá uma intensidade de luz proporcional ao sinal elétrico. Esta, por sua vez, é rececionada pela base do foto-transístor conduzindo a uma passagem de corrente do coletor para o emissor (procedimento normal do transístor bipolar). Todavia, o sistema pode revelar-se sensível nas transições de estado do transístor e, conseqüentemente revelar demasiada sensibilidade para a aplicação onde está inserido. Esta, porém, pode ser controlada através da conexão da base do transístor (representado pelo terminal 6) à massa recorrendo à utilização de uma resistência de valor *Ohmico* ( $\Omega$ ) elevada e viabilizando, desta forma, a estabilidade de comutação do foto-transístor. Verifica-se, maioritariamente, o emprego desta tipologia de sistemas de comunicação de corrente em circuitos separados.

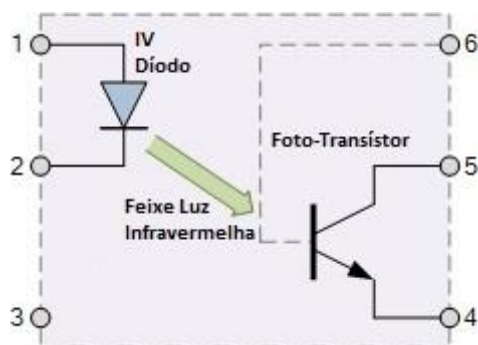


Figura 21 – Optoacopladores

### 3.3.6. CONTROLO DE MOTORES

Atendendo à importância dos motores utilizados no transporte e rotação da garrafa, tornou-se imprescindível a recorrência a um circuito eletrónico que objetive a sua configuração no que concerne à sua ordem de marcha, sentido e velocidade.

Visando o projeto proposto, a seleção dos motores a utilizar recaiu na utilização de motores *Direct Current* (DC). A revisão da literatura denota como melhor forma de controlo de motor (DC) a utilização de Pontes H, ([8]). Consideradas como circuitos simples, e como demonstrado na Figura 22, estas são constituídas por quatro elementos de comutação (usualmente transístores bipolares ou *Fiel-Effect Transistor* (FET) e *Insulated-Gate Bipolar Transistor* (IGBT), dispostas na configuração de H, surgindo numa posição central a carga – motor. O controlo dos seus transístores pode ser realizado de forma independente possibilitando a alternância do estado de condução entre ON e OFF e, conseqüentemente a passagem de corrente pelo motor - quer no sentido direto quer no inverso, controlando assim o seu sentido de rotação.

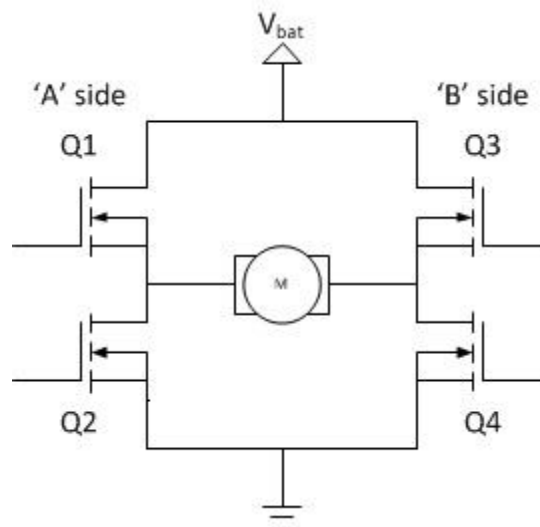


Figura 22 – Ponte H

Considerando a forma básica no que concerne à utilização de Pontes H e segundo a Figura 23, com a ativação dos transístores Q1 e Q4 é permitida a conexão do terminal esquerdo da carga à alimentação  $V_{bat}$  e, conseqüente ligação do terminal direito à massa. Estes processos facilitam o fluir da corrente pelo motor e energizando-o no sentido de rotação demonstrado na Figura estudada.

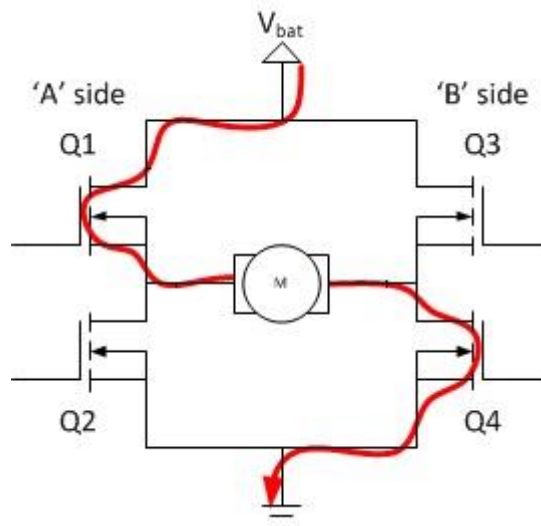


Figura 23 – Ponte H: Exemplo I

Visando a inversão do sentido de rotação anteriormente explanado torna-se necessária a desativação dos transístores seleccionados - Q1 e Q4, e a ativação dos inversos Q2 e Q3, procedendo assim à ligação do terminal direito a  $V_{bat}$  e o esquerdo à massa, como denotado na Figura 24.

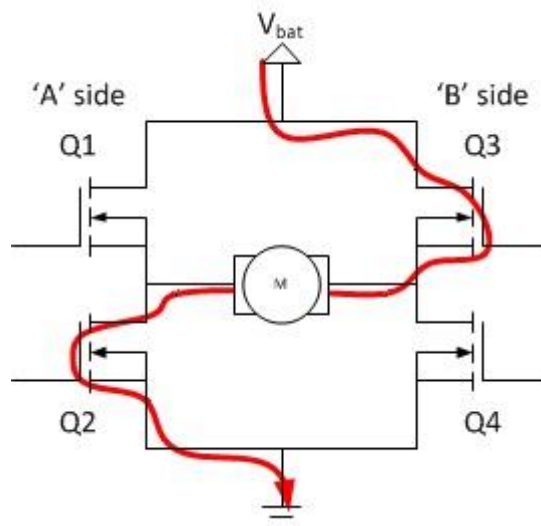


Figura 24 – Ponte H Exemplo II

Importa ressaltar que a ligação no mesmo espaço temporal dos transístores Q1 e Q2 ou Q3 e Q4 deve ser evitada pois, uma vez criada, a ligação entre  $V_{bat}$  e a massa torna-se inevitável, conduzindo à formação de curto-circuito da fonte de alimentação e provocando consequentemente danos na Ponte H e nos componentes a esta ligados. A recorrência a estas

Pontes H pode ainda possibilitar o processo de travagem do movimento rotacional do motor a partir da ativação de Q1 e Q3 ou Q2 e Q4 permitindo, desta forma, a ligação dos terminais à alimentação ou à massa, respetivamente.

Considerando a presente abordagem adotada que considera a operação dos transístores como estática, importa salientar que o controlo destes transístores pode, também ele, ser executado através da recorrência a um sinal *Pulse Width Modulation – PWM*. Neste, opera-se com uma frequência variável cujo tempo em que o interruptor permanece ativo varia.

Atendendo ao descrito anteriormente, tendo como base condições de trabalho ideais, facilmente se depreende que podem surgir alguns obstáculos. Considerando a comutação dos transístores, verifica-se que no contexto esta comutação instantânea não ocorre pela diferença de tempos do comando de controlo. Torna-se assim necessário proceder à posição em modo OFF de todos os transístores utilizados, pelo que na presença deste modo se verifica que o *rotor* do motor continua a rodar devido à sua energia cinética. Com o intuito de dissipar a energia acumulada pelas bobines do motor, providenciando um trajeto para o efeito desejado, surgiu a necessidade de se proceder à adição de díodos à Ponte H, como demonstrado pela Figura 25. Na eventualidade deste trajeto não ser proporcionado através da implementação de Díodos, na passagem ao estado OFF é gerada, pelos motores, uma elevada energia eletromagnética que quando acumulada se aplica aos terminais dos transístores e, conseqüentemente conduz à ocorrência de danos.

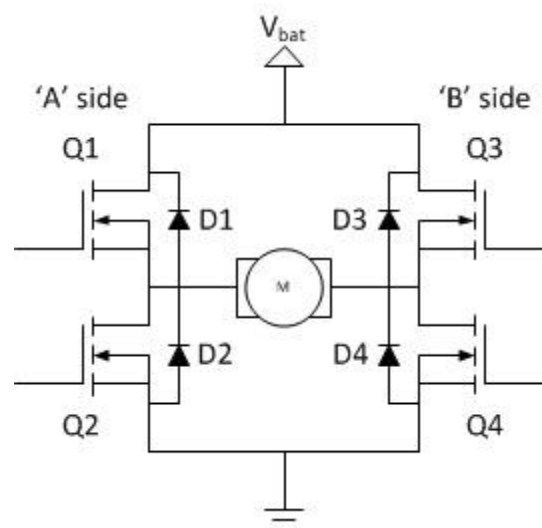


Figura 25 – Ponte H com Díodos

### **3.4. DESCRIÇÃO DE SOFTWARE**

Definido como a *fração lógica intangível de um computador e responsável pela manipulação, instrução de execução, redirecionamento e execução das atividades lógicas das máquinas*, o software assume particular relevância na construção e desenvolvimento do presente projeto.

#### **3.4.1. SISTEMA OPERATIVO RASPBERRY PI**

Definida como um sistema computacional de baixo custo com enorme variedade no que concerne às suas capacidades de trabalho, a Raspberry PI apresenta variada diversidade no que concerne aos sistemas operativos específicos. Os mesmos representam, maioritariamente distribuições Linux, a exemplificar Raspbian, Pidora, Openelec, Raspbmc e RiscOS.

- Raspbian – baseado em Debian Wheezy é um dos sistemas operativos mais utilizados;
- Pidora – baseado em Fedora Remix;
- Openelec – com foco em transformar a Raspberry Pi num servidor media center;
- Raspbmc – sistema operativo com mesmo foco do sistema anterior;
- RiscOS – com um kernel próprio destaca-se pela particularidade de não se basear em distribuição Linux ou Windows.

Considerando o sistema operativo *Raspbian* e atendendo ao seu número elevado de bibliotecas disponíveis, procedeu-se a sua seleção para o desenvolvimento do atual projeto.

#### **3.4.2. DESCRIÇÃO DO SOFTWARE UTILIZADO**

No que concerne à organização do software decidiu-se pela utilização de duas *threads* responsáveis quer pela interface gráfica disponível ao utilizador quer pelo controlo/leitura de motores/sensores e deteção do símbolo existente na garrafa.

Numa fase inicial procedeu-se ao desenvolvimento de um filtro que objetivasse a deteção exclusiva do símbolo pretendido. Tendo como referência o nível de protótipo, a seleção do símbolo visou ser casual permitindo que, em fases posteriores, o software desenvolvido para



a utilização do filtro anteriormente referido possibilite procedimentos como alterações do símbolo facilitando, desta forma, a sua adaptação às exigências impostas pelo mercado.

Atendendo ao software da máquina a desenvolver procedeu-se à sua subdivisão em duas partes: (1) Interface gráfica e o (2) Controlo. Considerando a primeira, Interface gráfica, determinou-se que o seu desenvolvimento será processado recorrendo à Plataforma PyQt, atendendo à sua capacidade de gerar um ficheiro .ui que, concomitantemente, contém o desenho de cada página do ambiente gráfico a utilizar, numa fase posterior, pelo software principal em Python. Todavia, e atendendo à parte de Controlo, recorrer-se-á à linguagem Python, responsável pelo controlo da totalidade de botões e mudanças de página. Atendendo ao software principal, procedeu-se à subdivisão em duas *threads*, responsáveis pelo controlo de motores, sensores e análise de imagem e uma outra pelo ambiente gráfico, respetivamente. Esta divisão garante a inexistência de bloqueio quer do ambiente gráfico quer do controlo da máquina aquando do processo de recolha de garrafas.

No presente projeto proceder-se-á à implementação de um controlo por software da totalidade de garrafas aceites e rejeitadas objetivando a obtenção de uma lista do número de garrafas. Os dados adquiridos permitirão a dedução de dados relevantes, a título de exemplo, o limite máximo de garrafas aceites e, desta forma, possibilitando, aquando da sua implementação, a intervenção de um técnico de manutenção.

## 4. IMPLEMENTAÇÃO

Após a definição da estrutura teórica que permitiu orientar o Projeto pretende-se, neste capítulo, abordar os procedimentos imprescindíveis para a implementação do mesmo. Desta forma apresentam-se os esquemas elétricos, o software desenvolvido bem como os cálculos executados.

### 4.1. IMPLEMENTAÇÃO DE HARDWARE

Objetivando a construção de um protótipo de testes, visou ser necessário um dimensionamento e construção da totalidade da componente elétrica ao nível de hardware. Como tal, e procurando a colocação dos detetores infravermelhas na melhor localização possível no protótipo a desenvolver, delineou-se a construção de placas de circuito impresso ou PCB, isto é, placas que suportam mecanicamente componentes eletrónicos que se encontram eletricamente ligados com auxílio de pistas condutoras. Tornou-se, todavia imprescindível o desenho das placas necessárias para o funcionamento do protótipo. As mesmas sofreram um processo de divisão em módulos originando placas distintas, entre as quais, uma placa principal de maiores dimensões e com capacidade de proceder ao controlo dos motores e regulação da tensão, bem como outras de menores dimensões que visam albergar quer detetores e emissores infravermelhos quer LED's de alto brilho, que objetivam uma melhoria significativa da iluminação do interior do protótipo. Com o intuito de explanar o dimensionamento de cada esquema elétrico bem como dos seus componentes, emerge o presente subcapítulo.

#### 4.1.1. DIMENSIONAMENTO DO ESQUEMA ELÉTRICO

Por forma a obter o melhor desempenho torna-se necessário calcular o valor de cada resistência presente em cada circuito elétrico. Desta forma este subcapítulo apresentará todos os cálculos elaborados.

#### 4.1.1.1. Emissor Infravermelho

A obtenção do valor das resistências para cada LED emissor de infravermelhos é realizada através da recorrência a datasheet – documento que apresenta de forma discriminada os dados e características do equipamento, objetivando a aquisição de dados como *queda de tensão* do mesmo e *valor de corrente*, seguidamente apresentados, com o intuito de deter o seu funcionamento na sua potência ideal

Queda de tensão LED: 1.2V

Corrente LED: 20mA

Considerando uma alimentação de 5V, como demonstrado na Figura 26, procede-se ao cálculo das resistências para cada LED, onde **i** representa a corrente que percorre o circuito, **R** o valor da resistência, **VR** o valor da queda de tensão da resistência, **VLED** identifica a queda de tensão do LED e **VCC** define a queda de tensão do conjunto resistência-LED.

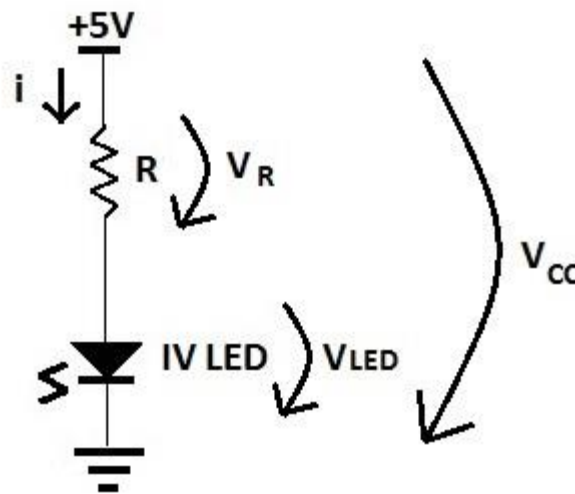


Figura 26 - Emissor de Infravermelhos

Considerando a Lei de Ohm, que define que a constante de proporcionalidade entre U e I seria a «resistência elétrica», normalmente definido por R, recorreu-se à sua utilização objetivando determinar o valor da resistência para que o LED funcione na sua melhor *performance*.

$$V_R = V_{CC} - V_{LED} \quad (10)$$

$$R = \frac{V_R}{i} \quad (11)$$

Tendo como queda de tensão na resistência um valor de  $V_R = 3,8V$  equação 10, torna-se possível concluir que o valor da resistência será de  $R = 190\Omega$ , equação 11.

#### 4.1.1.2. Detetor Infravermelho

Considerando o fototransistor utilizado no presente projeto e atendendo às suas características descritas no Datasheet, foi possível a obtenção da corrente de coletor de  $0.2mA$  e a queda de tensão coletor-emissor de  $0.8V$  no estado de saturação. A obtenção do estado anteriormente mencionado requer uma incidência luminosa cujo valor de potência assume-se de  $1mW/cm^2$ . Encontrados os parâmetros necessários para a aquisição do valor da resistência *pull down*, representada na Figura 27, realizaram-se os cálculos imediatamente explanados.

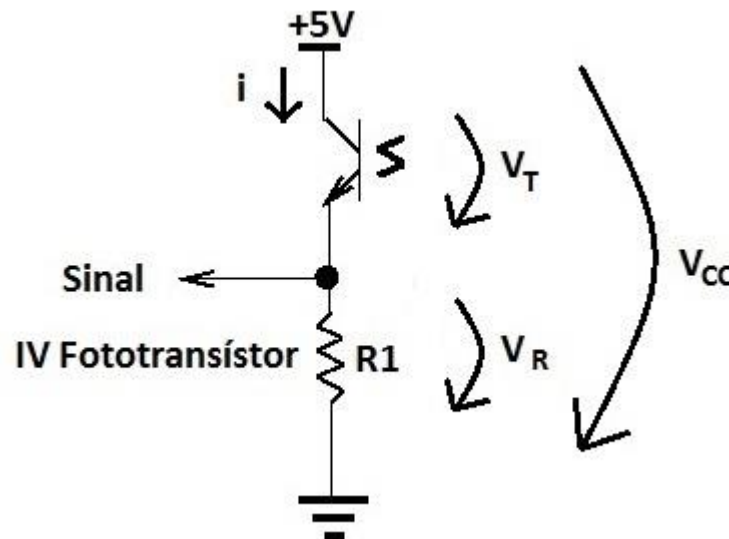


Figura 27 – Detetor de Infravermelhos

$$V_R = V_{CC} - V_T \quad (12)$$

$$R = \frac{V_R}{i} \quad (13)$$

Tendo como queda de tensão na resistência um valor de  $V_R = 4,2V$  equação 12, torna-se possível concluir que o valor da resistência será de  $R = 21k\Omega$ , equação 13.

Com o intuito de regular a sensibilidade de leitura do fototransistor, poder-se-ia recorrer à utilização de um potenciômetro cuja ação se caracteriza pelo auxílio no ajuste da resistência.

#### 4.1.1.3. LEDs de Alto Brilho

Objetivando a melhoria significativa da luminosidade do interior da máquina recorreu-se à implementação de três LEDs brancos de alto brilho. Desta forma, e atendendo uma vez mais às características específicas dos mesmos, procedeu-se ao cálculo da resistência necessária para uma série de três LEDs, como exemplificado na Figura 28.

Queda de tensão do LED: 3.2V

Corrente LED: 20mA

Tensão de alimentação: 12V

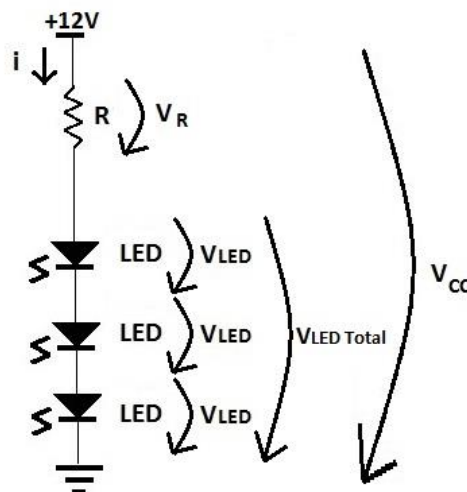


Figura 28 – Cálculo LEDs Alto Brilho

$$V_{LED\ Total} = 3 \times V_{LED} \quad (14)$$

$$V_R = V_{CC} - V_{LED\ Total} \quad (15)$$

$$R = \frac{V_R}{i} \quad (16)$$

Multiplicando a queda de tensão de cada LED pela totalidade dos LEDs presentes na série obtém-se  $V_{LED\ Total} = 9,6V$  equação 14, sendo assim possível calcular a queda de tensão na

resistência  $V_R = 2,4V$  equação 15, com todos estes dados e recorrendo à lei de Ohm conclui-se que o valor da resistência será de  $R = 120\Omega$ , equação 16.

#### 4.1.2. ESQUEMA ELÉTRICO E PLACA PCB PRINCIPAL

A elaboração do esquema elétrico principal, explanada na Figura 29, acarreta componentes eletrónicos essenciais e responsáveis pela regulação de tensão, controlo de motores bem como portas de comunicação (ficha multipólo) essenciais para o envio e receção dos sinais de e para o microcomputador *Raspberry Pi*.

Atendendo à alimentação total do circuito da placa PCB principal, denota-se que a mesma apresenta na sua constituição dois terminais “*Main Power*” permitindo que a mesma seja alimentada por uma fonte externa. No que concerne à **tensão de entrada**, fornecida por uma fonte DC externa, denota-se que a mesma pode variar entre 7V a 25V, consequência dos reguladores de tensão presentes no circuito elétrico.

Após o fornecimento desta alimentação geral ao circuito assume-se a necessidade de se proceder à regulação do mesmo a 5V recorrendo, contudo, à instalação de dois reguladores evitando a sobrecarga e, desta forma, possibilitando quer a alimentação da totalidade dos componentes eletrónicos presentes na Placa quer uma porta USB, por sua vez responsável pela alimentação da Raspberry Pi e ecrã. Considerando a alimentação dos motores e dos LEDs de alto brilho, a regulação da tensão de entrada foi estabelecida a 12V. Nestes procedeu-se, tal como anteriormente, à colocação de dois reguladores, responsável pela alimentação dos Motores e pelos LEDs, respetivamente.

Visando os pontos de ligação entre **PCB principal - PCBs modulares** (referentes aos emissores e detetores de infravermelhos) recorreu-se à utilização de dois terminais duplos e dois triplos de parafuso. Os primeiros, terminais duplos, foram utilizados atendendo a que as PCBs referentes aos **emissores** de infravermelhos requerem unicamente uma alimentação de 5V, provenientes do regulador de tensão. Por sua vez, verifica-se que as PCBs referentes aos **detetores** de infravermelhos requerem não só uma alimentação de 5V como também um terceiro terminal que permita o envio do sinal proveniente do detetor recorrendo-se, desta forma, a terminais triplos.

Considerando a Raspberry Pi e atendendo a que as suas entradas digitais funcionam com uma voltagem de 3.3V, surgiu a necessidade de se proceder à conversão do sinal proveniente

dos detetores infravermelhos para 3.3V. Considerando as diversificadas possibilidades existentes, a decisão recaiu na utilização de optoacopladores pela garantia de proteção da Raspberry Pi na ocorrência de dano do mesmo. Apresentando no seu interior um díodo emissor, ativado pelo sinal proveniente do detetor de infravermelhos, os optoacopladores permitem, através do transístor o fluir da corrente proveniente do terminal de 3.3V da Raspberry Pi.

No que concerne ao controlo da alternância dos LEDs de alto brilho selecionou-se um transístor bipolar de junção que visa assumir-se como sendo a «chave» de alimentação do circuito dos LEDs. O controlo do transístor é realizado através de um optoacoplador que procede ao isolamento do sinal proveniente da Raspberry Pi do sinal de controlo do transístor. Por sua vez, a conexão PCB-LEDs de Alto Brilho é realizada através de dois terminais de parafuso.

A PCB principal apresenta ainda na sua constituição um controlador de motores, L293 – Ponte H, capaz de fornecer aos motores utilizados tensões entre 4.5V e 36V a 1A. Com o intuito de tornar o sistema mais modular determinou-se a conexão de dois terminais duplos de parafuso à saída da ponte H para cada motor, possibilitando assim a alternância de motor de forma mais simples e rápida. Imediatamente unido a estes terminais encontram-se díodos de *freewheeling* uma vez inexistentes internamente nas pontes H utilizadas. Importa ainda ressaltar, e mantendo sempre a ideologia de um sistema modular, que todos os sinais de controlo da Ponte H utilizada provêm da Raspberry Pi pelo que foi determinada a implementação de uma ficha multipólo DB15, denominada por X1, que possibilita verificar a correspondência dos primeiros dois terminais com as entradas digitais na Raspberry, por sua vez detentoras dos sinais digitais dos detetores de infravermelhos.

Esta ficha contém, respetivamente, nos terminais três e quatro os potenciais GND e 3.3V, provenientes da Raspberry e que objetivam a alimentação de todo o circuito de entradas digitais da mesma. O potencial GND, anteriormente descrito, visa a implementação do mesmo referencial quer no circuito elétrico externo quer na Raspberry. Por sua vez, e atendendo aos terminais entre o 5 e 10, é possível encontrar as saídas digitais da Raspberry responsáveis pelo controlo da Ponte H.

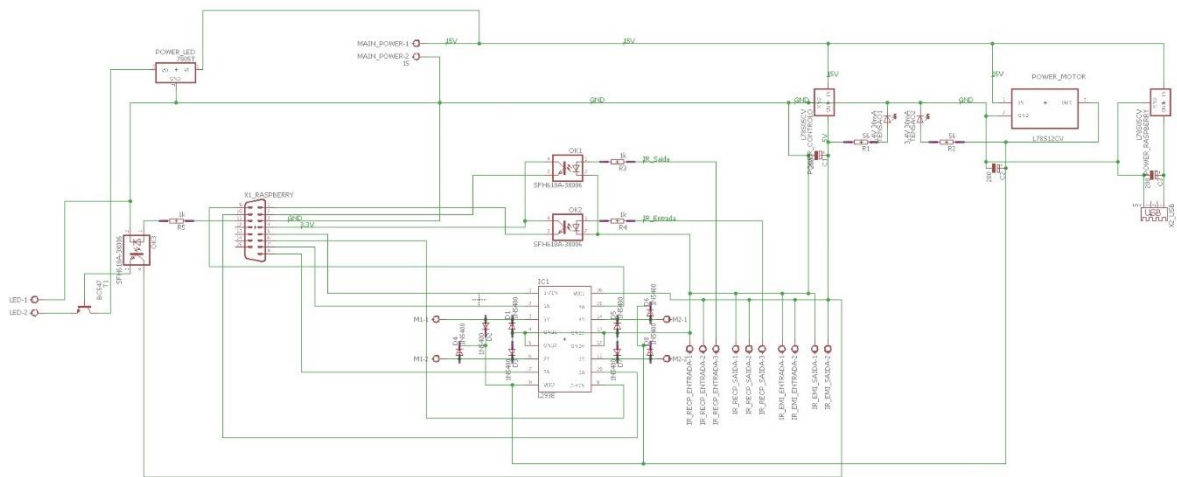


Figura 29 – Esquema Elétrico Geral

Considerando a sua complexidade de circuito, a PCB principal apresenta ainda pistas em ambas as faces da placa que permitirão, num contexto de desenvolvimento prático, a conexão dos seus componentes. Estas encontram-se explanadas na Figura 30 e na Figura 31.

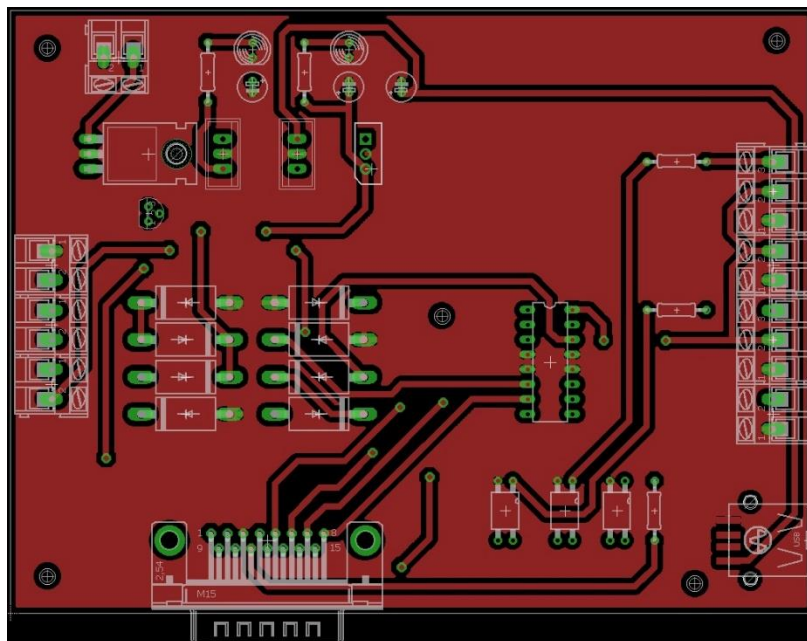


Figura 30 – PCB Geral Face Superior



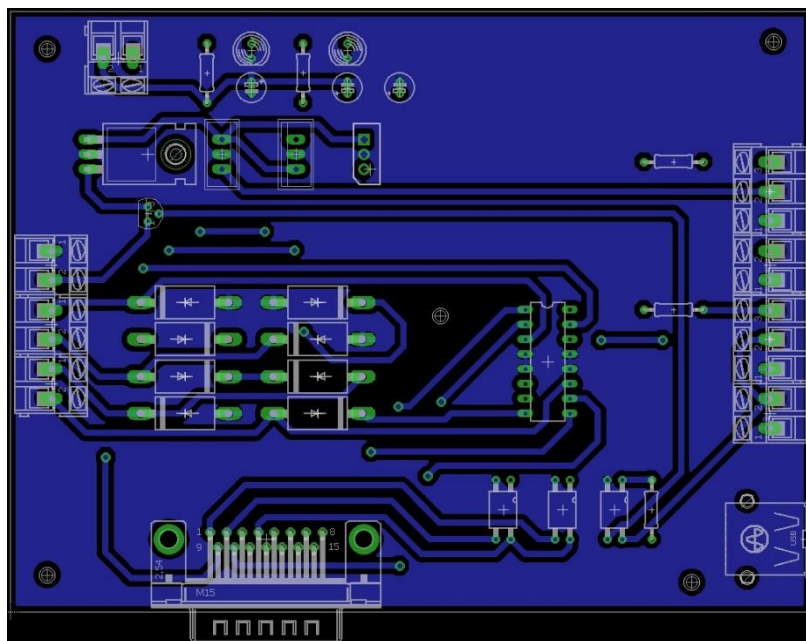


Figura 31 – PCB Geral Face Inferior

#### 4.1.3. PCBs MODULARES

Visando o desenvolvimento de um projeto maioritariamente modular procedeu-se à construção de PCBs de pequenas dimensões, num total de cinco, duas com um emissor, duas com detetor infravermelho e uma última com LEDs de alo brilho.

##### 4.1.3.1. Detetor infravermelho

Atendendo aos detetores infravermelhos foram delineados três terminais, dos quais dois confinados para a alimentação e um outro para o sinal proveniente do fototransístor, explicitado na Figura 32. Por sua vez, a Placa PCB em causa apresenta ainda na sua constituição uma resistência de *Pull down* (21k $\Omega$ ) e um fototransístor, como possível observar na Figura 33.

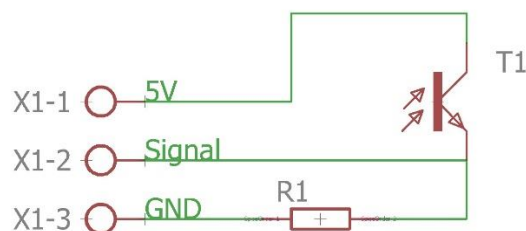


Figura 32 – Esquema Elétrico do Detetor Infravermelho

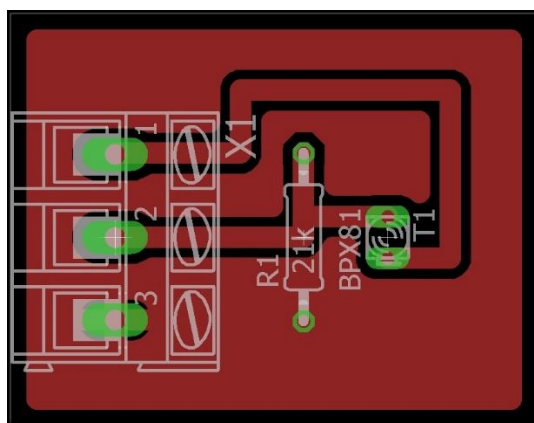


Figura 33 – PCB Detetor Infravermelho

#### 4.1.3.2. Emissor Infravermelho

Considerando os emissores infravermelhos e o desenvolvimento do seu esquema elétrico, optou-se por se proceder à colocação de terminais de alimentação, uma resistência de valor anteriormente calculado ( $190\Omega$ ) bem como o LED emissor infravermelho, como representado na Figura 34 e Figura 35.

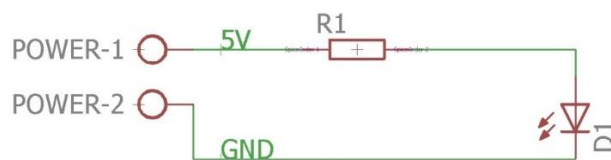


Figura 34 – Esquema Elétrico Emissor Infravermelho

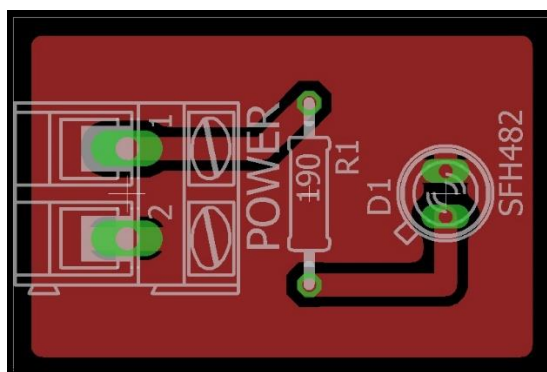


Figura 35 – PCB Emissor Infravermelho

#### 4.1.3.3. LEDs de Alto Brilho

Visando a melhoria da qualidade luminosa no interior da máquina e objetivando uma melhoria da análise de imagem recorreu-se à utilização dos LEDs brancos de alto brilho, como anteriormente mencionado. Em concordância com o descrito foi desenvolvido o esquema elétrico dos mesmos, utilizando um paralelo de três LED em série como apresentado na Figura 36. Procedeu-se, conseqüentemente ao desenho da respectiva PCB utilizando uma resistência para cada série e dois terminais para a alimentação, como visível na Figura 37. Optou-se por se proceder ao desenvolvimento desta PCB de forma isolada, tendo por base a Placa principal, uma vez se assumir necessário o ajustamento da posição da mesma para se atingir as condições luminosas ideais.

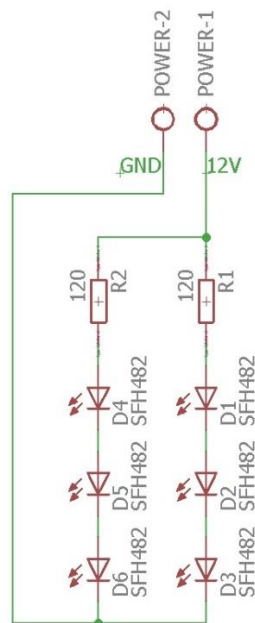


Figura 36 – Esquema Elétrico LEDs Alto Brilho

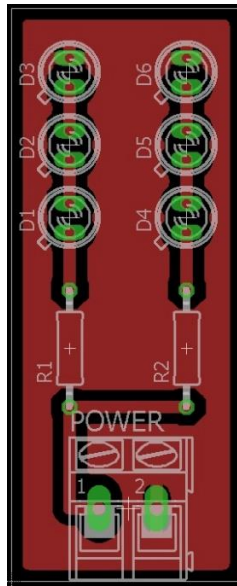


Figura 37 – PCB LEDs Alto Brilho

A produção de PCBs de menores dimensões requer uma alimentação com tensão correta, objetivando a não utilização de equipamentos de regulação de tensão.

Findado o dimensionamento no que concerne ao hardware, emerge a possibilidade de construção do protótipo, explanado na Figura 38, que permita a realização de testes e validação do software a desenvolver.

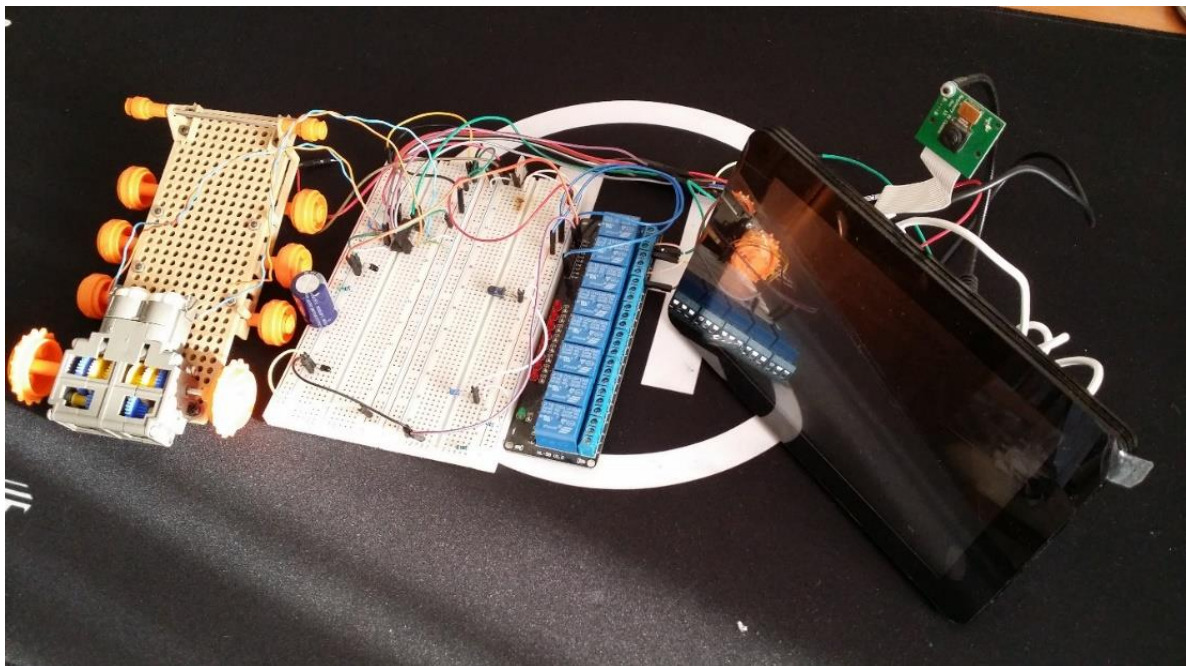


Figura 38 – Protótipo de testes

## 4.2. IMPLEMENTAÇÃO DE SOFTWARE

Definindo Software como a *sequência de instruções a serem interpretadas por um computador com o objetivo de executar tarefas específicas*, o presente subcapítulo expõe o software utilizado no presente projeto visando a garantia de um funcionamento uníssono do sistema. Será ainda explanada a totalidade das dependências a instalar, o percurso efetuado para a criação do filtro responsável pela análise de imagem bem como o desenvolvimento da interface gráfica responsável pelo controlo do protótipo em desenvolvimento.

### 4.2.1. INSTALAÇÃO DAS DEPENDÊNCIAS NA RASPBERRY PI

Objetivando a construção de um sistema capaz de proceder a atividades como análise de imagem, controlo de saídas e entradas digitais da Raspberry Pi e gestão de ambiente gráfico tornou-se essencial a instalação de dependências como OpenCV, Python, Kivy e QtCreator. Neste âmbito, o processo teve início com a atualização do sistema operativo *Raspbian* conjuntamente com a instalação de pacotes de leitura e escrita de imagens em formatos como *Joint Photographic Experts Group* (JPEG), *Portable Network Graphics* (PNG) e *Tagged Image File Format* (TIFF) bem como de um pacote de leitura e escrita de vídeo, objetivando a integração/carregamento de vídeos com OpenCV. Concluída a instalação dos sistemas anteriormente mencionados, procede-se à Instalação da Livraria OpenCV concomitantemente com a ferramenta de desenvolvimento Python.

O processo de instalação da Livraria OpenCV é iniciado através do download do ficheiro compactado zip existente no website ([6])

Realizado o download e descompactação do ficheiro procede-se à compilação da Livraria OpenCV recorrendo ao comando *make*. Importa ressaltar que a versão de Open CV utilizada no presente projeto, 3.0.0, embora não sendo mais recente, reúne as capacidades de deteção necessárias ao desenvolvimento do protótipo garantindo ainda suporte para a versão Python. Assim, e de encontro ao anteriormente descrito, procedeu-se à implementação e instalação da versão 3.4 da linguagem de programação Python, recorrendo ao comando:

```
“apt-get install python3.4-dev”
```

Terminada a instalação de Python procedeu-se ainda à instalação de um pacote básico desta linguagem, **NumPy**. Permitindo o trabalho com matrizes multidimensionais e arrays, a

linguagem em causa permite que estes últimos, emergidos da ligação existente OpenCV-Python, sejam passíveis de conjecturar.

Concluídas as instalações de OpenCV e Python e certificado o seu correto funcionamento, procede-se à continuidade do processo através da instalação de ferramentas necessárias para o desenvolvimento gráfico. Inicialmente procedeu-se à instalação da biblioteca Kivy objetivando o desenvolvimento de ambientes gráficos e, conseqüentemente, a interface homem-máquina. Todavia, e atendendo às desvantagens apresentadas por esta biblioteca, das quais se destaca, elevado tempo a disponibilizar para o seu desenvolvimento, a necessidade de se proceder ao desenho de cada botão, caixa de texto e/ou imagem a sua utilização foi inviabilizada. Em concordância com o descrito previamente surge a necessidade de seleccionar uma nova ferramenta de desenvolvimento, opção que recaiu sobre Qt Creator com *Integrated Development Environment* (IDE) e exemplificado na Figura 39. A seleção desta ferramenta assumiu-se como proveitosa uma vez permitir um desenvolvimento mais rápido e intuitivo.

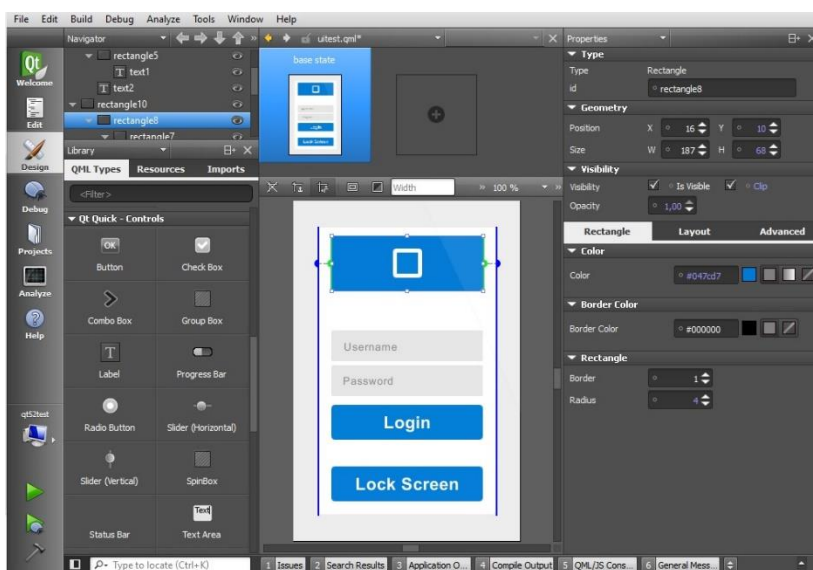


Figura 39 – Qt Creator IDE

#### 4.2.2. CRIAÇÃO DE FILTRO PARA ANÁLISE DE IMAGEM

Objetivando a deteção e reconhecimento do símbolo presente na garrafa pelo protótipo a desenvolver emergiu a necessidade de se proceder à criação de um filtro utilizado, numa fase posterior para a identificação do símbolo. Desta forma, e em concordância com o descrito anteriormente, a sua criação terá como base de desenvolvimento a cascata de Haar que

preconiza a aquisição de imagens positivas e negativas. Considerando positivas as imagens cujo símbolo a detetar se encontra presente importa especificar, numa fase posterior, a região onde o mesmo se encontra. Todavia, no que concerne às imagens negativas, que podem apresentar temática diversificada, o símbolo a detetar verifica-se inexistente.

Concretizada a seleção do símbolo a utilizar, representado na Figura 40, e da sua submissão ao redimensionamento para 50x50 bem como da sua colocação nas garrafas, recorre-se ao mesmo para se proceder ao treinamento.



Figura 40 – Símbolo a detetar

Todavia, emergiu a necessidade de, após a seleção do símbolo a utilizar, se dispor de diversificadas imagens escolhidas de forma casual e de designação negativa. Essa necessidade visou ser conducente do desenvolvimento de um programa baseado na Linguagem Python e presente no Anexo A. O mesmo recorre a um website detentor de diversificadas imagens passíveis de serem descarregadas procedendo, de forma automática, à sua transformação nomeadamente da cor (em tonalidades de cinza), do seu redimensionamento (para o tamanho de 100x100 pixéis) bem como do seu renomear (através da partícula “neg/”, da atribuição de um número (iniciado em 0) e do formato jpg). Desta forma, obtém-se de forma direta as imagens prontas a utilizar.

Porém, o facto de o programa anteriormente mencionado recorrer a um website conduz a que, ocasionalmente, as imagens deixem de estar disponíveis no servidor online. Assim, e atendendo à realidade expressa, é automaticamente processada e descarregada uma imagem que apresenta intrinsecamente a explicação do ocorrido colocando em vogue a necessidade de se proceder ao desenvolvimento de um programa, em Python e presente no Anexo B, capaz de detetar as imagens em causa e proceder à sua eliminação.

O processo de eliminação das imagens é realizado através da função OpenCV *imread*, que procede à leitura da imagem que se pretende eliminar e procede ao carregamento da imagem negativa com igual função utilizada procedendo à comparação de ambas as imagens. Na eventualidade das mesmas se assumirem iguais procede-se à eliminação do diretório das imagens negativas. Este processo repete-se até que a totalidade das imagens negativas sejam analisadas.

Visando a criação de um ficheiro de texto com o diretório da totalidade de imagens negativas em formato de lista, com denominação *bg.txt*, e posteriormente utilizado pela Biblioteca OpenCV, emergiu ainda a necessidade de se proceder ao desenvolvimento de um outro programa em Python, presente no Anexo C.

Os processos desenvolvidos até ao momento permitiram deter unicamente as imagens negativas emergindo, neste momento, a necessidade de se criar imagens positivas detentoras do símbolo selecionado. Consequentemente, e recorrendo às potencialidades da Biblioteca OpenCV procede-se à sobreposição do símbolo nas imagens negativas gerando, desta forma, imagens positivas. Este processo é executado através da função *opencv\_createsample*, providenciada por OpenCV, que funde o símbolo com a imagem negativa e gera uma nova imagem. A cada imagem criada, o símbolo é colocado em diferentes ângulos e rotações aumentando consideravelmente a capacidade de deteção. Paralelamente ao processo mencionado é produzido um ficheiro de texto *info.lst* que contém, para além da denominação da imagem, a localização e dimensão do símbolo (x, y, largura, altura). Adquirido o ficheiro anteriormente descrito, emerge a possibilidade de se proceder à criação de um novo ficheiro *vec*, detentor das amostras positivas utilizadas e posteriormente, utilizadas no treinamento.

Concluídas as diferentes etapas do processo encontram-se reunidas as condições necessárias ao treino da cascata. Atendendo à importância da função *opencv\_traincascade* no processo de treino importa ressaltar a importância de preencher determinados parâmetros, nomeadamente a localização onde será guardado o treino da cascata (*-data*), bem como a localização dos ficheiros anteriormente criados como o ficheiro *vec* e o ficheiro *bg.txt* (*-bg*). Verifica-se ainda a necessidade de especificar o número de imagens positivas e negativas (*-numPos*, *-numNeg*) a utilizar em cada estágio de treino da cascata, bem como o número de estágios pretendidos (*-numStages*). Realizadas diversificadas interações, verificam-se que os valores que obtiveram melhores resultados na deteção do símbolo foram ***-numPos: 2200***, ***numNeg: 1500*** e ***numStages: 12***.



Findado o treino da cascata é concebido um ficheiro denominado *cascade.xml*, ficheiro esse utilizado no programa Python para a deteção do símbolo presente na garrafa e presente numa localização especificada.

#### 4.2.3. DESENVOLVIMENTO DA INTERFACE GRÁFICA

Designada como a *forma de interação entre o usuário e um programa desenvolvido*, o progresso da interface gráfica determinou, numa fase inicial, como base de desenvolvimento a Biblioteca Kivy. Recorrendo à sua tecnologia procedeu-se ao desenho da primeira página do mesmo, explanada na Figura 41.

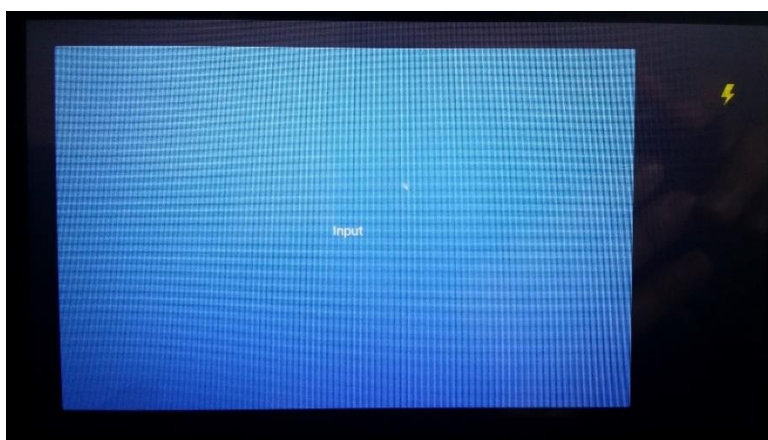


Figura 41 – GUI Desenvolvida em Kivy

Não obstante, e atendendo à complexidade exigida pela aplicação cujo processo impõe o desenvolvimento de códigos específicos para ações de menor sofisticação, a título de exemplo o desenho de um botão, a sua utilização denotou ser bastante morosa, emergindo a necessidade de se proceder à seleção de uma aplicação diferenciada. Em concordância com o descrito, procedeu-se à seleção do software QtCreator visando maior celeridade no processo de desenvolvimento da aplicação.

Recorrendo ao software selecionado, QtCreator, foram formuladas sete páginas diferentes, ficheiro *.ui* presente no Anexo F. A **página 0** – Inicial demonstrada na Figura 42, estabelece o primeiro contacto entre máquina-utilizador apresentando, unicamente um botão que iniciará todo o processo assim como o logotipo do protótipo que pode ser adaptado, num contexto futuro, a um cliente ou empresa.

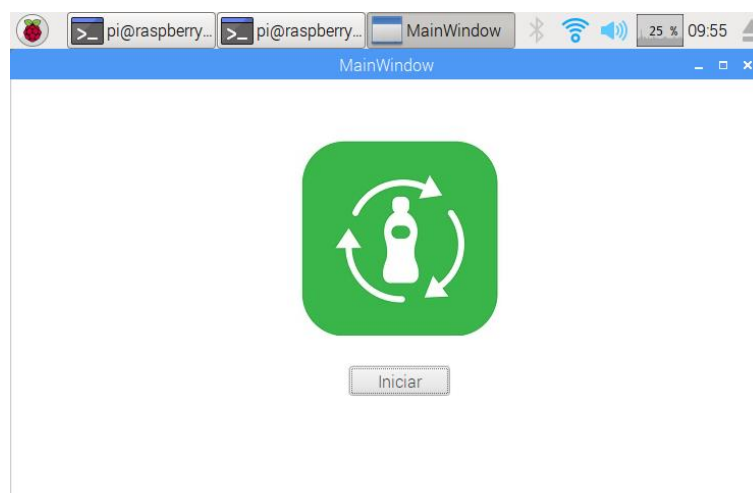


Figura 42 – Ambiente Gráfico Página 0 – Inicial

Iniciado o processo e objetivando a sua continuidade através da introdução de uma garrafa no sistema, surge a **página 1** – Recolha, que visa a compreensão pelo utilizador da ação em desenvolvimento pela máquina – processamento da garrafa introduzida, como é possível observar na Figura 43.

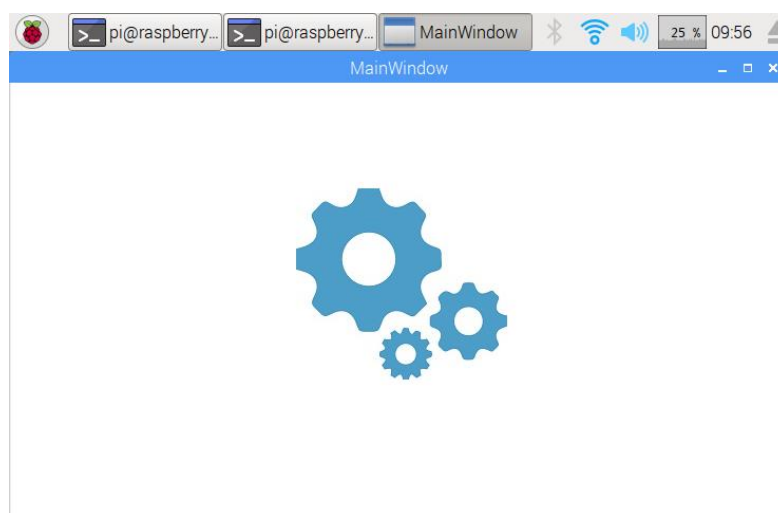


Figura 43 – Ambiente Gráfico Página 1 – Recolha

Concluído o ciclo de recolha é exposta a **Página 2** – Contabilização de Garrafas, identificada na Figura 44, responsável pela transmissão de informações referentes ao número total de garrafas aceites e rejeitadas, procedentes desde o início do ciclo inaugurado pelo utilizador

na Página 0 – Inicial. Contudo, na eventualidade do contentor de armazenamento se encontrar cheio, o redireccionamento ocorrerá para a **página 6** – Valor Retorno Cheio, apresentada na Figura 47.



Figura 44 – Ambiente Gráfico Página 2 – Contabilização de Garrafas

Esta página coloca o utilizador perante duas escolhas: (1) iniciar novamente o ciclo através da introdução de uma nova garrafa e consequentemente passando para a **página 1** - Recolha ou (2) terminar o processo e prosseguir para uma nova página, de nomenclatura **Página 3** – Valor Retorno, exposta na Figura 45.



Figura 45 – Ambiente Gráfico Página 3 – Valor Retorno

Atendendo à Página 3 verifica-se que se encontra explanado o número total de garrafas aceites multiplicado pelo valor atribuído a cada uma. Através da presente fórmula matemática torna-se possível identificar o valor a receber, pelo utilizador, referente à retoma pelas suas garrafas.

Objetivando a retribuição do valor referente às garrafas introduzidas, é providenciada a impressão de um talão como perceptível na **Página4** - Impressão, explanada na Figura 46.

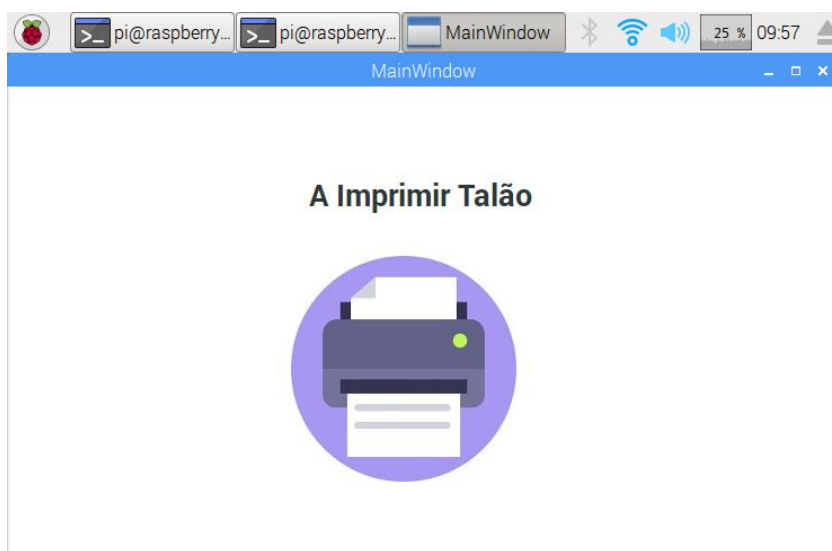


Figura 46 – Ambiente Gráfico Página 4 – Impressão

Na eventualidade de ser atingido o valor total da capacidade do contentor, o utilizador depara-se com a Página 6 – Valor Retorno Cheio, como na Figura 47, onde a par do que é apresentado na Página 3 é acrescentada a informação de que o contentor se encontra cheio. A partir deste momento só é dada ao utilizador a opção de impressão do talão. Sendo posteriormente redirecionado para a Página 5 – Manutenção, apresentada na Figura 48.

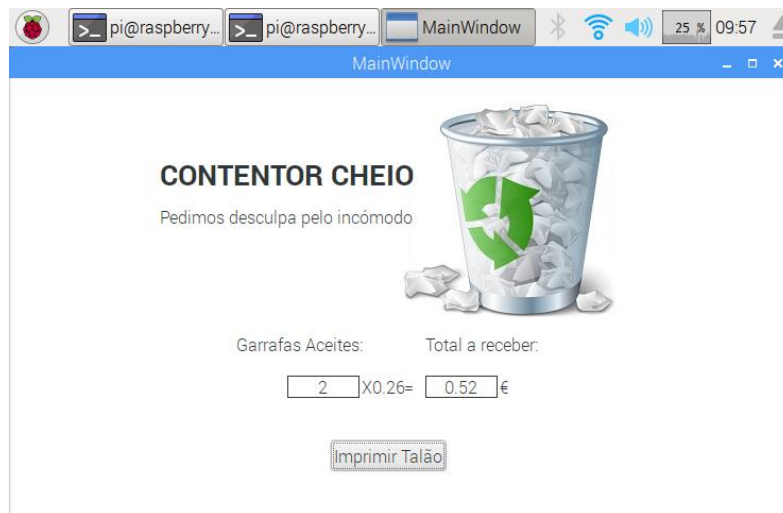


Figura 47 – Ambiente Gráfico Página 6 – Valor Retorno Cheio

Desta forma, após a impressão do talão anteriormente referido, a máquina adquire o modo de manutenção visível com a criação da Página 5 - Manutenção e exposto na Figura 48, que inibe a utilização desta até ao seu esvaziamento.

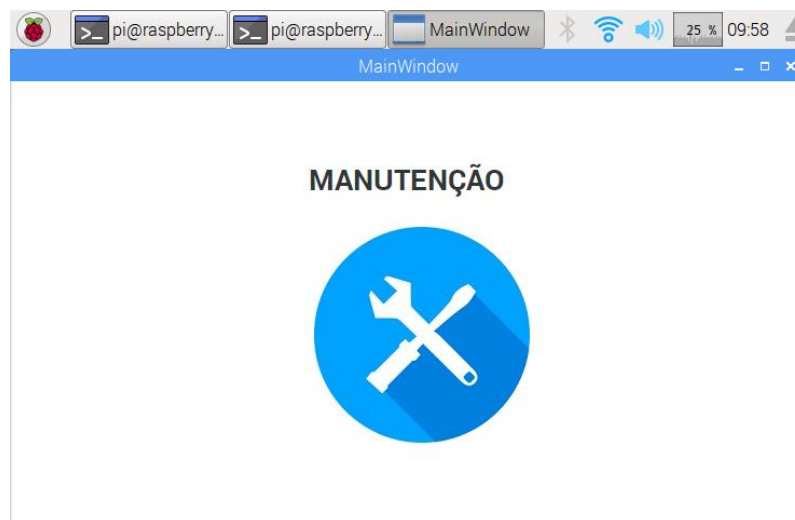


Figura 48 – Ambiente Gráfico Página 5 – Manutenção

Na eventualidade da totalidade das garrafas introduzidas não corresponderem aos parâmetros anteriormente delineados, o utilizador será novamente direcionado para a página inicial, denominada de Página 0 – Inicial, objetivando o reiniciar de todo o projeto.

Objetivando uma melhor compreensão do encadeamento de páginas do ambiente gráfico é apresentado um Fluxograma na Figura 49.

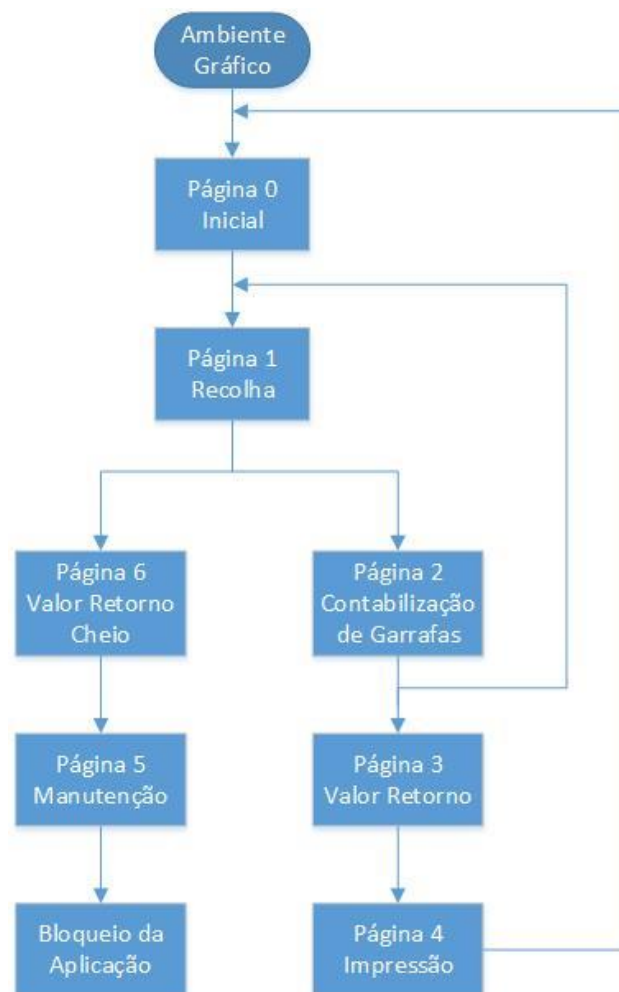


Figura 49 – Fluxograma Ambiente Gráfico

#### 4.2.4. SOFTWARE GERAL

Considerando a complexidade do protótipo, o software desenvolvido, presente no anexo E, suportou diversificadas interações. Numa fase inicial, a gestão da interface gráfica ocorria simultaneamente ao controlo do processo de recolha e análise da garrafa. Contudo, e verificadas incontáveis dificuldades, nomeadamente durante a fase de testes (como o bloqueio da aplicação gráfica pela sobreposição de ações pelo software), procedeu-se à criação de duas *threads*: (1) responsável pelo controlo do processo e análise de imagem e (2) responsável pelo ambiente gráfico e interface com o utilizador, como possível observar na Figura 50. A tipologia de construção de software utilizada conduziu à necessidade evidente de serem efetuadas comunicações das variáveis entre as duas *threads*. Desta forma, e atendendo a esta necessidade eminente procedeu-se à utilização de variáveis globais – definidas como variável “Iniciar” e “Página”, que permitiram a partilha da variável entre as *threads*. A variável “Iniciar” sofre o seu processo de atualização pela *thread* GUI

desencadeando, consequentemente, o ciclo da *thread* Processo para a recolha da garrafa. Por sua vez, a variável Página é atualizada pela *thread* de Processo e lida pela *thread* GUI que, respeitando o valor guardado nessa variável, procede ao carregamento da respetiva página.

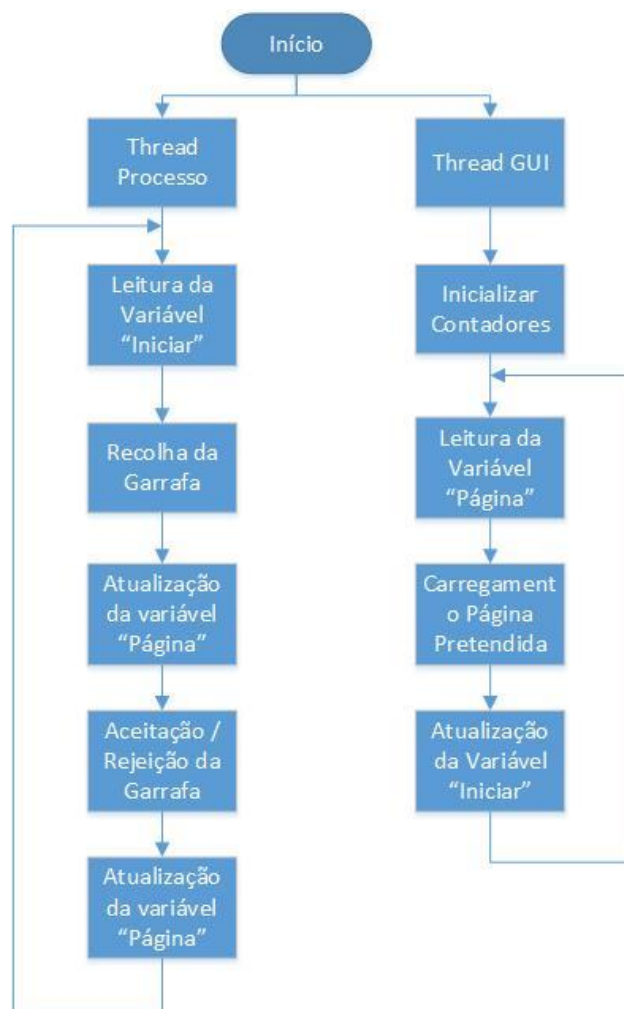


Figura 50 – Fluxograma Geral

Visando o estudo discriminado da thread de processo procedeu-se à elaboração do fluxograma Figura 51.

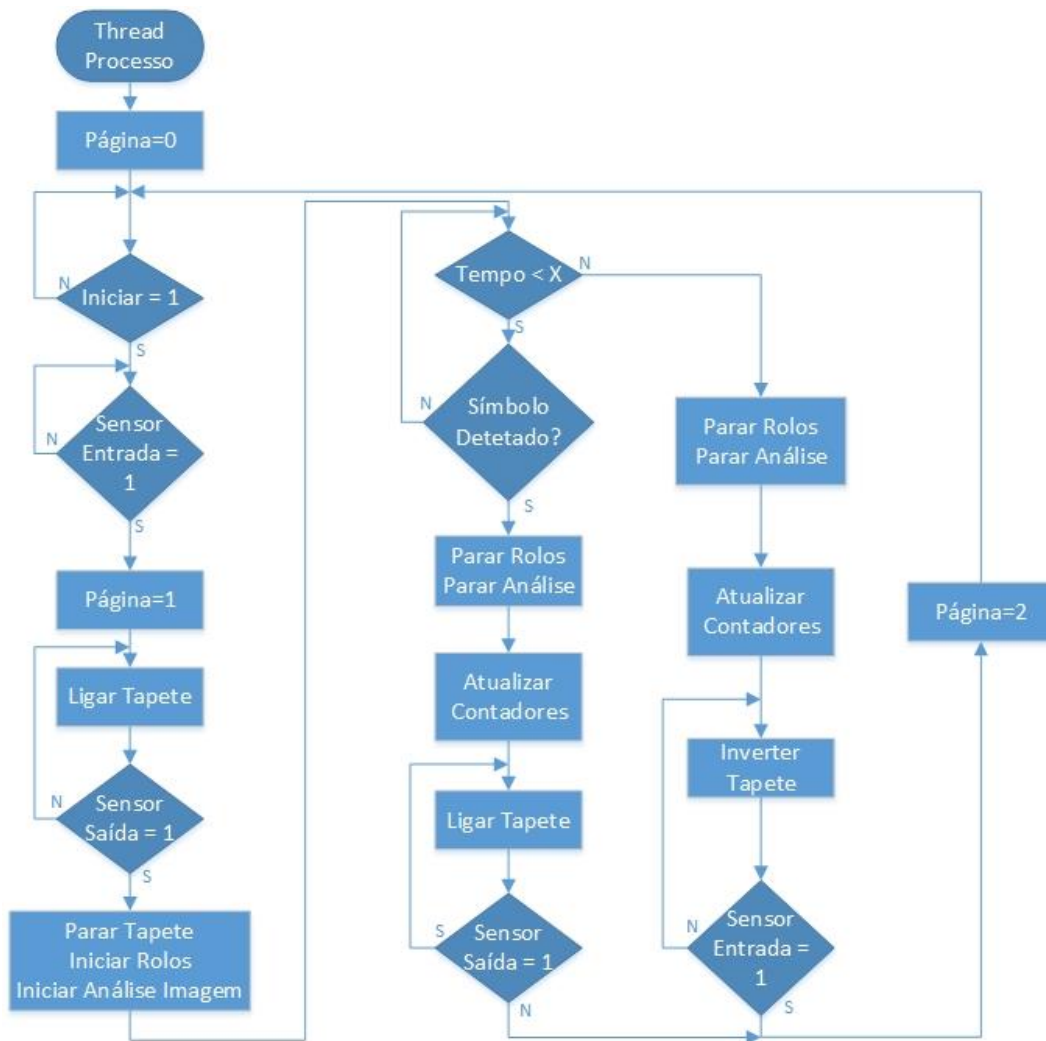


Figura 51 – Fluxograma Ciclo Processo

Considerada a primeira *thread* como a responsável pelo controlo do processo e análise de imagem importa realçar, como primeiro acontecimento, o carregamento do valor zero para a variável *página* visando, desta forma, o carregamento da página numerada como zero. Objetivando o início do processo, desencadeado pelo premir do botão, desenvolveu-se uma condição que mantém o ciclo de processo em modo *aguardar*, prevendo uma alteração na variável de início do ciclo. Assumida como verdadeira a condição anterior, é aguardado pelo sistema a deteção do objeto. Esta por sua vez é executada através dos sensores infravermelhos de entrada que ativam, consequentemente, após a alteração da página para 1, o motor responsável pelo movimento do tapete (sentido direto).

Visando a continuidade do processo, o sistema requer um período de pausa que preconiza que o objeto a analisar proceda à interrupção do feixe de infravermelhos do sensor de saída



parando o tapete e permitindo a análise do objeto na posição correta para o efeito. Uma vez adquirida a posição de análise de imagem o motor responsável pela rotação dos rolos e consequentemente pela rotação da garrafa é ativado, iniciando o processo de análise de imagem.

Delineado, numa fase inicial, para ser processado em cinco segundos – tempo passível de alteração em Python, e na eventualidade de **não detetar o símbolo selecionado**, o processo de análise de imagem sofre uma interrupção que é procedida pela paragem dos rolos e consequente atualização dos contadores.

Divididos em duas grandes vertentes, é possível encontrar (1) contadores de garrafas aceites e rejeitadas locais e (2) contadores de garrafas aceites e rejeitadas gerais. Os primeiros correspondem aos contadores de cada cliente no qual, e após o término da inserção de garrafas, é realizado um reset. Por sua vez, os contadores de garrafas aceites e rejeitadas gerais incluem a totalidade destas mesmas garrafas, de manutenção em manutenção, gravando um ficheiro *.txt* na raiz do sistema e permitindo, de forma mais fácil, o acesso por parte da manutenção.

Concluído o processo de atualização dos contadores a garrafa é devolvida ao utilizador da máquina através da inversão do sentido de rotação dos motores do tapete e terminando no momento em que a garrafa interrompe o feixe de luz do par infravermelho de entrada. Desta forma, e visando findar o processo, procede-se à atualização da variável página para 2.

Na eventualidade do símbolo selecionado ser encontrado num tempo inferior aos cinco segundos anteriormente mencionados, a análise e rotação dos rolos é cessada exatamente no momento da deteção procedendo-se, automaticamente, à atualização das variáveis. Consequentemente é ordenado o arranque do motor do tapete (no sentido direto) que cessa aquando da perda de deteção da garrafa pelo sensor de infravermelhos de saída – momento em que a mesma é enviada para o depósito. Em concordância com o anteriormente descrito, e realizada uma atualização da variável página.

Consideradas as possibilidades apresentadas importa ressaltar que o sistema retorna a sua fase inicial, aguardando a ordem de início proveniente da *thread* responsável pelo ambiente gráfico.

Atendendo à segunda *thread* referente ao ciclo *Graphical user interface* (GUI) verifica-se que a mesma é responsável pela análise do valor da variável *página*, atualizado na *thread* Processo, e consequente apresentação da página pretendida. Com o intuito de perceber o seu funcionamento é apresentado o seu fluxograma na Figura 52.

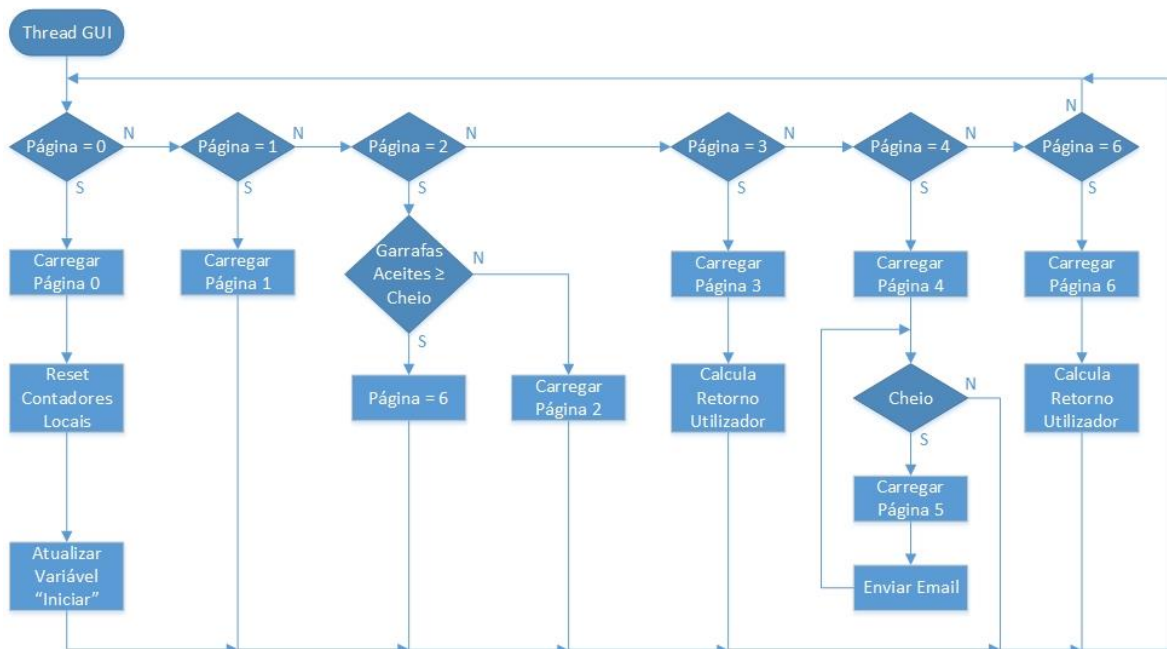


Figura 52 – Fluxograma Ciclo GUI

Na eventualidade de a página selecionada a apresentar ser a de designação zero é executado o seu carregamento seguindo-se o reset dos contadores locais. Por sua vez, e visando a apresentação do valor um (1) na variável é apresentada a página 1. Considerado o valor dois (2) e, atuando anteriormente à alteração de página, importa considerar se o número total de garrafas aceites visa ser superior ao número de garrafas permitidas a armazenar, sendo carregada a página 6 na eventualidade de verificada a hipótese ou, caso não seja verificada procede-se ao carregamento normal da página 2 pretendida pelo ciclo. Caso o valor pretendido na variável seja três (3) é carregada a página correspondente, procedendo ainda ao cálculo do valor a retornar ao utilizador tendo como base o número de garrafas aceites e o valor comercial correspondente a cada garrafa. Este, por sua vez, é carregado para as respetivas caixas de texto presentes na página em questão. Visando a atribuição do valor quatro (4) à variável é, consequentemente carregada a página 4. Considerando a hipótese de o contentor de garrafas se encontrar cheio é carregada a página cinco (5), e mantida até à intervenção pela manutenção. Paralelamente é enviado um email, para um endereço a

configurar, com informações relativas ao número e morada do local onde a máquina se encontra instalada bem como o número de garrafas existentes no contentor.

Na eventualidade de ser solicitada a página seis (6) é processado o cálculo do valor de retorno como anteriormente explicado na variável de valor 3.

Concebidas as duas threads anteriormente mencionadas, verifica-se que os seus ciclos permanecem, até ao encerramento da aplicação, continuamente em *loop*. A presente funcionalidade emerge da necessidade de se proceder à conceção de uma aplicação cuja intervenção deva ser exclusivamente realizada por um técnico de manutenção, que procederá à interrupção da aplicação recorrendo ao comando *Ctrl + c*.

## 5. TESTES E RESULTADOS

Considerada como uma das vertentes fundamentais do presente projeto, a análise de imagem desenvolvida procurou uma contínua precisão e exatidão objetivando a melhor detecção exequível, patente com uma menor percentagem de erro.

Desta forma, e concordantemente com o descrito, procedeu-se à realização de quatro treinos da cascata de classificadores todos eles diversificados com variações referentes ao número de imagens positivas e negativas e número de estágios. A realização dos diversificados treinos possibilitou a obtenção de diferentes resultados, buscando o melhor ficheiro de treino para a máquina em questão.

Como explanado na Tabela 2, foram realizados quatro treinos usando um número distinto de imagens positivas e negativas. O primeiro e segundo treino caracterizam-se pela recorrência a 1700 imagens positivas e 1000 negativas. Todavia, para o terceiro e quarto treino foram empregues 3700 imagens positivas e 2500 negativas. Em ambos os casos, é apenas notória uma variação do número de estágios que se denota de 10 e 15 estágios.

**Tabela 2 Treinos Realizados**

	<b>Primeiro treino:</b>	<b>Segundo treino:</b>	<b>Terceiro treino:</b>	<b>Quarto treino:</b>
<b>Número de Imagens Positivas</b>	1700	1700	3700	3700
<b>Número de Imagens Negativas</b>	1000	1000	2500	2500
<b>Número de Estágios</b>	10	15	10	15

Num outro prisma, e atendendo às configurações entre treinos são conferidos diversificados tempos de treinamento. A título de exemplo, a recorrência a uma máquina com processador i7 e 8GB de memória RAM apresenta tempos díspares. Tratando-se ainda de um processo moroso como possível averiguar na seguinte Tabela 3 – Tempo de treinamento.

**Tabela 3 Tempo de treinamento**

Número do Treino	Tempo de execução	Rácio de Aceitação
1	1h 46min	0,0030
2	3h 30min	0,0002
3	6h 39min	0,0023
4	10h 36min	0,0001

Após a realização dos testes aos diferentes treinos é possível deduzir algumas conclusões. Atendendo ao ficheiro referente ao primeiro treino, e sendo este caracterizado por um menor número de imagens e estágios, verifica-se que o sistema ostenta maior imprecisão na ação de deteção passível de percecionarmos quer através do rácio de aceitação, num valor de 0.0013 – valor fornecido na fase final de cada treino, quer através da realização de testes práticos. Torna-se assim possível percecionarmos a presença de inúmeros falsos positivos.

Para além da deteção do símbolo pretendido, também diferentes parâmetros ou diversificados símbolos do rótulo são captados. Por sua vez, e considerando o ficheiro relativo ao segundo treino, denota-se um aumento considerável da deteção - num rácio de aceitação de 0,0002 (consideravelmente inferior ao obtido no primeiro treino) paralelamente ao aumento do tempo utilizado para a finalidade pretendida.

No que concerne ao terceiro e quarto treino perceciona-se um valor de rácio de deteção idêntico ao obtido nos treinos 1 e 2 pelo que é possível deduzir que o aumento do número de imagens visa não influenciar de forma notória o resultado obtido. Desta forma, denota-se que o aumento de estágios visa uma maior precisão na deteção do símbolo pretendido. Assente no princípio descrito, verificam-se resultados semelhantes nos treinos 1 e 3, bem como no 2 e 4.

Considerando os resultados obtidos procedeu-se à realização de um novo treino que agrupasse configurações intermédias dos treinos 1 e 4 anteriormente mencionados, como é possível observar na seguinte Tabela 4 – Treino Seleccionado.

**Tabela 4 Treino Seleccionado**

Número de Imagens Positivas	2200
Número de Imagens Negativas	1500
Número de Estágios	12

O treino anteriormente explanado, com durabilidade de três horas e vinte e nove minutos (3h 29min), visou a obtenção de um rácio de aceitação de 0,0007 descobrindo, desta forma, um equilíbrio entre as vertentes velocidade-deteção e erro assumindo-se, até ao momento, como a melhor opção encontrada.

No que consiste ao desenvolvimento do ambiente gráfico e realçando a eleição do IDE QtCreator como melhor opção, importa realçar o seu perfeito funcionamento ao longo de todo o projeto, destacando ainda as vantagens emergidas pela sua utilização, nomeadamente a facilidade de desenvolvimento da mesma e a sua expansão. Porém, a mesma acarreta, unicamente como desvantagem, a sua limitação no que concerne ao desenvolvimento de uma aplicação atrativa em termos visuais.

No que concerne ao software principal importa realçar os diversificados ajustamentos processados que conduziram à sua obtenção como apresentado no presente projeto e onde a totalidade das funcionalidades pretendidas apresentam um funcionamento total.

Como descrito em maior pormenor no capítulo quatro, o software apresentado encontra-se dividido em duas *threads* que permitiram manter o ambiente gráfico paralelamente à execução do processo de recolha e de análise de imagem.

Atendendo à conceção do ficheiro responsável pela contabilização do número de garrafas aceites e rejeitadas, verifica-se que a sua gravação é procedida no formato *.txt*, facilitando a leitura do mesmo no processo de manutenção. O ficheiro em causa apresenta finalidades variadas, destacando a utilização dos dados obtidos para a elaboração, num contexto futuro,

de estatísticas relativas ao número de clientes, número de garrafas aceitas e rejeitadas e até dos locais onde as máquinas apresentam maior utilização.

Considerando o símbolo, e delimitando as suas dimensões a 50 pixels por 50, em garrafas de 0.33L verificou-se, através da realização de diversificados testes experimentais, que com as dimensões em causa se obteve um resultado concordante com o pretendido.

Considerando o hardware utilizado, importa ressaltar que a produção das PCBs foi realizada com recorrência a um *Computer Numerical Control* (CNC). O presente processo assumiu-se como proveitoso uma vez que a impressão de um circuito de dupla face com as dimensões preconizadas teria um custo superior. Realizados diversificados testes, o sistema em causa apresentou elevada fiabilidade e robustez.

Executados os diversificados testes denota-se que o protótipo desenvolvido atingiu os objetivos delineados numa fase inicial procedendo, de forma correta, à deteção do símbolo configurado durante a fase do treino de cascata de classificadores, não procedendo à obtenção de falsos positivos.

Por sua vez, e no que concerne ao ambiente gráfico do protótipo, verifica-se o seu funcionamento pleno mesmo aquando do funcionamento da função processo cuja necessidade de recursos de processamento é superior, como o processo de análise da garrafa. Este assume-se ainda como intuitivo e de fácil utilização por qualquer utilizador.

Atendendo à deteção e movimentação da garrafa, verifica-se que o seu dimensionamento foi devidamente realizado uma vez trabalharem sem a ocorrência de erros.

## 6. CONCLUSÃO

Considerada a crescente industrialização mundial e consequente incremento do consumo de materiais como o plástico, são incrementadas medidas conducentes de maior responsabilidade social no que concerne à gestão e proteção ambiental. Aludindo ao conteúdo anteriormente explanado facilmente se depreende a urgente necessidade de criar medidas que visem salvaguardar o ambiente.

Neste sentido foi proposta a construção de um protótipo capaz de analisar e proceder à recolha de garrafas de plástico com critérios criados/planeados. Os critérios impostos para a recolha de garrafas centram-se na identificação de um símbolo previamente selecionado e presente no rótulo da garrafa que permita, consequentemente e numa fase posterior, a sua aceitação e recolha pelo protótipo desenvolvido bem como o seu encaminhamento para centros de reciclagem.

Todavia, e objetivando o contínuo sucesso do protótipo a desenvolver, foram efetuados diversificados estudos de mercado visando identificar e perceber as tecnologias emergentes bem como, através de uma revisão da literatura, os processos requeridos para o estudo da análise de imagem em tempo real ou os softwares e linguagens de programação que melhor se ajustavam aos objetivos previamente delineados.

Em concordância com a revisão da literatura efetuada percebeu-se que a oferta tecnológica em vogue permite a construção de máquinas extremamente eficientes, mas consequentemente caras. Contrariando esta tendência, e através da utilização de componentes simplistas como sensores de imagem, consola tátil e Raspberry Pi, procedeu-se à construção da máquina proposta. Todavia, e evidenciando a importância do software para a construção do protótipo proposto, importa denotar o seu desenvolvimento. Ostentando como base teórica a Cascata de Haar e o seu treinamento para a obtenção do melhor filtro, atendendo à relação velocidade/erro, conjuntamente com a gestão da totalidade do ambiente gráfico e o processo de recolha e análise da garrafa, o software desenvolvido visa uma utilização intuitiva e fácil do protótipo. O presente projeto exigiu ainda desenvolvimentos



no campo do hardware, destacados através da realização de inúmeros cálculos, desenho e construção de esquemas elétricos e PCBs.

Relativamente aos objetivos preconizados e com os recursos existentes deduz-se o seu alcance, na generalidade. Foram atingidos com êxito pretendido, o estudo das soluções do mercado atual, o estudo da análise de imagem, do software QtCreator, de Linguagens de Programação e velocidade dos mecanismos, bem como a diminuição de custos de produção e a definição dos requisitos necessários para o desenvolvimento da máquina. O objetivo construção da máquina capaz de proceder à análise e recolha de garrafas atendendo aos critérios exigidos, anuncia um menor grau de consecução relativamente aos anteriores, em parte pela dificuldade na vertente mecânica. Todavia, e objetivando ultrapassar a dificuldade emergida procedeu-se à construção de um protótipo capaz de proceder à realização de testes que atestam o funcionamento do mesmo.

Inerente à realização deste projeto, foram identificados um conjunto de dificuldades e obstáculos dos quais se destacam o treino da cascata de classificadores e a sua integração no software, a seleção da Biblioteca responsável pela criação do ambiente gráfico – iniciado em Kivy e concretizado através do IDE QtCreator bem como o englobar da gestão do processo de recolha da garrafa paralelamente à gestão do ambiente gráfico.

Pretende-se, anteriormente à conclusão, enfatizar algumas melhorias a introduzir como um novo modo de retribuição do valor da garrafa ao cliente, a introdução de um leitor de código de barras e aumentar a tipologia de garrafas a aceitar – como aceitação de novos materiais, permitindo não só atender à problemática do consumo excessivo de plástico, mas atender a um sistema de seleção e separação dos materiais.

Aproveito também, para manifestar a minha satisfação quanto à orientação conferida pelo docente orientador, na medida em que se prontificou, sempre que necessário, a responder clara e objetivamente às questões colocadas. Assim, a referida orientação possibilitou a atenuação de alguns obstáculos supramencionados.

Em suma, a exigência elencada ao desenvolvimento do projeto permitiu um crescimento pessoal e profissional apresentado pelo elencar de ferramentas que visam não só a gratificação pessoal como a contribuição para uma melhor Engenharia Eletrotécnica de Computadores.

## Referências

- [1] **Bayer Filter** - [Consult. 18 fev. 2017]. Disponível em WWW:<URL:<http://abertoatedemadrugada.com/2011/09/o-que-e-o-padrao-bayer.html>>.
- [2] BRADSKI, Gary *et al.* - Learning OpenCV. 2008.
- [3] **Camera Board - Raspberry Pi** - [Consult. 6 mar. 2017]. Disponível em WWW:<URL:<https://www.raspberrypi.org/blog/creating-the-camera-board/>>.
- [4] **CCD vs. CMOS** - [Consult. 12 fev. 2017]. Disponível em WWW:<URL:<https://www.teledynedalsa.com/imaging/knowledge-center/appnotes/ccd-vs-cmos/>>.
- [5] DAMAS, Luís; - **Linguagem C**. 10ª ed. Rio de Janeiro : [s.n.]
- [6] **Download OpenCV from SourceForge.net** - [Consult. 27 set. 2017]. Disponível em WWW:<URL:<https://sourceforge.net/projects/opencvlibrary/files/opencv-unix/3.0.0/opencv-3.0.0.zip/download>>.
- [7] FREUND, Yoav; SCHAPIRE-, Robert E. - A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting\*. **Journal of Computer and System Sciences SS1504 journal of computer and system sciences**. 55:1997.
- [8] **H-Bridges** - [Consult. 2 ago. 2017]. Disponível em WWW:<URL:<http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>>.
- [9] **Inovação Tecnológica : TOMRA** - [Consult. 24 jan. 2017]. Disponível em WWW:<URL:<https://www.tomra.com/pt/collection/reverse-vending/tomra-technology/>>.
- [10] JACKSON, Cody - Learning to Program Using Python. 2013.
- [11] LUÍS, Autor .; PATSKO, Fernando - TUTORIAL Aplicações, Funcionamento e Utilização de Sensores. 2006).
- [12] MARENGONI, Maurício; STRINGHINI, Stringhini - Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista de Informática Teórica e Aplicada**. . ISSN 21752745. 16:1 (2010) 125–160.

- [13] **Raspberry DSI** - [Consult. 7 mar. 2017]. Disponível em WWW:<URL:[https://www.petervis.com/Raspberry\\_Pi/Raspberry\\_Pi\\_LCD/Raspberry\\_Pi\\_LCD\\_DSI\\_Display\\_Connector.html](https://www.petervis.com/Raspberry_Pi/Raspberry_Pi_LCD/Raspberry_Pi_LCD_DSI_Display_Connector.html)>.
- [14] **Raspberry Pi Hardware** - [Consult. 23 fev. 2017]. Disponível em WWW:<URL:<https://www.raspberrypi.org/documentation/hardware/raspberrypi/>>.
- [15] **Understanding How a Voltage Regulator Works** - [Consult. 26 set. 2017]. Disponível em WWW:<URL:<http://www.analog.com/en/technical-articles/how-voltage-regulator-works.html>>.
- [16] VIOLA, Paul; JONES, Michael J. - Robust Real-Time Face Detection. **International Journal of Computer Vision**. 57:2 (2004) 137–154.

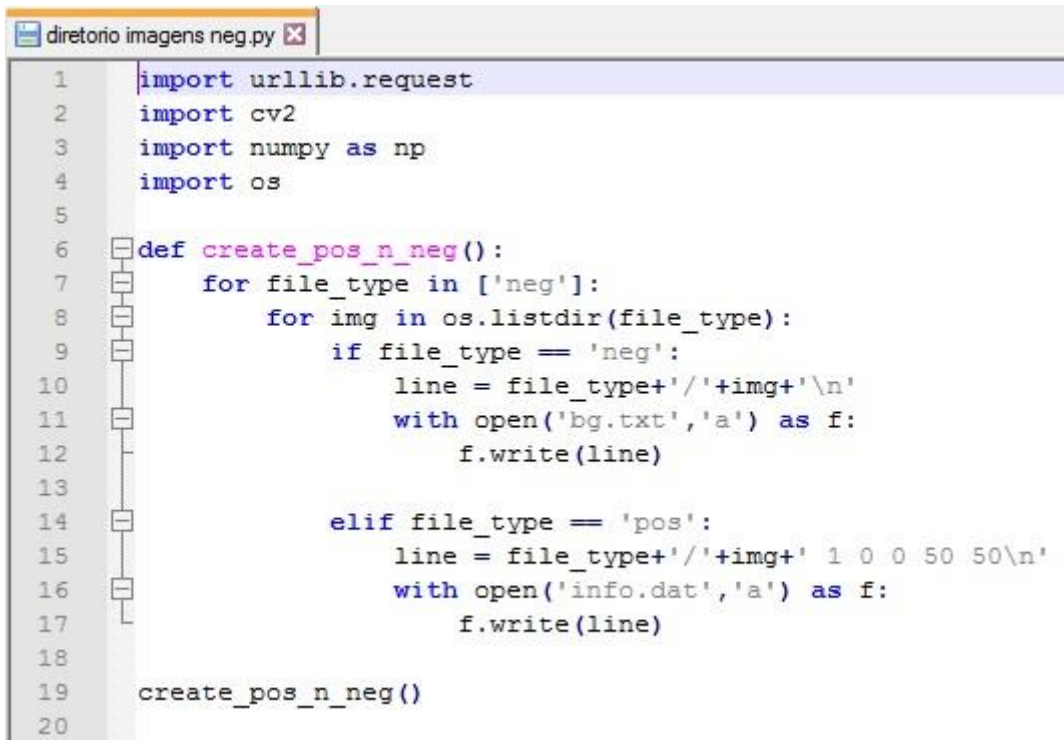
## Anexo A. Download Imagens Negativas

```
Download_Images.py
1  import urllib.request
2  import cv2
3  import numpy as np
4  import os
5
6  def download_images():
7      neg_images_link = 'http://image-net.org/api/text/imagenet.synset.geturls?wnid=n03245889'
8      neg_image_urls = urllib.request.urlopen(neg_images_link).read().decode()
9
10     if not os.path.exists('neg'):
11         os.makedirs('neg')
12
13     pic_num = 0
14
15     for i in neg_image_urls.split('\n'):
16         try:
17             urllib.request.urlretrieve(i, "neg/"+str(pic_num)+'.jpg')
18             img = cv2.imread("neg/"+str(pic_num)+'.jpg', cv2.IMREAD_GRAYSCALE)
19             resized_image = cv2.resize(img, (100,100))
20             cv2.imwrite("neg/"+str(pic_num)+'.jpg', resized_image)
21             pic_num += 1
22
23         except Exception as e:
24             print(str(e))
25
26     download_images()
27
```

## Anexo B. Filtragem Imagens Negativas

```
eliminar_imagens.py
1 import urllib.request
2 import cv2
3 import numpy as np
4 import os
5
6 def eliminar_imagens():
7     for file_type in ['neg']:
8         for img in os.listdir(file_type):
9             for ugly in os.listdir('uglies'):
10                 try:
11                     current_image_path = str(file_type)+'/'+str(img)
12                     ugly = cv2.imread('uglies/'+str(ugly))
13                     question = cv2.imread(current_image_path)
14
15                     if ugly.shape == question.shape and not(np.bitwise_xor(ugly,question).any()):
16                         os.remove(current_image_path)
17
18                 except Exception as e:
19                     print(str(e))
20
21 eliminar_imagens()
22
```

## Anexo C. Criação do ficheiro bg.txt



```
1 import urllib.request
2 import cv2
3 import numpy as np
4 import os
5
6 def create_pos_n_neg():
7     for file_type in ['neg']:
8         for img in os.listdir(file_type):
9             if file_type == 'neg':
10                 line = file_type+'/'+img+'\n'
11                 with open('bg.txt','a') as f:
12                     f.write(line)
13
14             elif file_type == 'pos':
15                 line = file_type+'/'+img+' 1 0 0 50 50\n'
16                 with open('info.dat','a') as f:
17                     f.write(line)
18
19 create_pos_n_neg()
20
```

## Anexo D. Tempo de Análise de Imagem

```
Tempo_detecao.py
1 from picamera.array import PiRGBArray
2 from picamera import PiCamera
3 import numpy as np
4 import cv2
5 import time
6
7 camera = PiCamera()
8 camera.resolution = (640, 480)
9 camera.framerate = 32
10 rawCapture = PiRGBArray(camera, size=(640, 480))
11
12 symbol_cascade = cv2.CascadeClassifier('cascade_1_ensaio.xml')
13
14 time.sleep(0.1)
15
16 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
17     last_time = time.perf_counter()
18     image = frame.array
19     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
20     symbol = symbol_cascade.detectMultiScale(gray, 1.3, 5)
21     print(time.perf_counter()-last_time)
22     for (x,y,w,h) in symbol:
23         cv2.rectangle(image, (x,y), (x+w, y+h), (255,255,0), 2)
24         cv2.imshow("Frame", image)
25
26     key = cv2.waitKey(1) & 0xFF
27
28     rawCapture.truncate(0)
29
30 if key == ord("q"):
31     break
32
```

## Anexo E. Programa Principal

```
ProcessPython_funcional_email.py x
1  # coding=utf-8
2  from PyQt5 import QtCore, QtWidgets, uic
3  from picamera.array import PiRGBArray
4  from picamera import PiCamera
5  from multiprocessing import Queue
6  import numpy as np
7  import cv2
8  import time
9  import RPi.GPIO as GPIO
10 import sys
11 import threading
12 import decimal
13 import csv
14 import smtplib
15 from email.mime.text import MIMEText
16
17 qtCreatorFile = "mainwindow.ui"
18 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
19
20 class MyApp(QtWidgets.QMainWindow, Ui_MainWindow, threading.Thread):
21     global pagina
22     global d
23     global valor_garrafa
24     global n_cheio
25     global tempo_analise
26     global para
27     global assunto
28
29     valor_garrafa = 0.26 ## Valor em euros
30     n_cheio = 3 ## Unidades
31     tempo_analise = 5 ## Valor em segundos
32     para = "pedrodanny@gmail.com"
33     assunto = 'RVM 1'
34
35     pagina = 0
36     d = Queue()
37     def __init__(self):
38         QtWidgets.QMainWindow.__init__(self)
39         Ui_MainWindow.__init__(self)
40         self.setupUi(self)
41         self.stackedWidget.setCurrentIndex(0)
42         thread1= threading.Thread(target=self.GUI)
43         thread2= threading.Thread(target=self.processo)
44         thread1.start()
45         thread2.start()
46
47     def GUI(self):
48         global garrafasR
49         garrafasR = 0
50         global resultado
51         resultado = 0.00
52         global garrafasA
53         garrafasA = 0
54         global garrafasAT
```

Python file



```
ProcessPython_funcional_email.py
54 global garrafasAT
55 global clicked
56 clicked = 0
57 global cheio
58 cheio = 0
59 teste_cheio = 0
60 flag_5 = 0
61 while True:
62     pagina = d.get()
63     if pagina == 0:
64         self.stackedWidget.setCurrentIndex(0)
65         garrafasA = 0
66         garrafasR = 0
67     elif pagina == 1:
68         self.stackedWidget.setCurrentIndex(1)
69     elif pagina == 2:
70         file = open("Garrafas.txt", "r")
71         teste_cheio = file.read(2)
72         file.close()
73         if int(teste_cheio) >= n_cheio:
74             print("teste", teste_cheio)
75             cheio = 1
76             pagina = 6
77             d.put(pagina)
78         if int(teste_cheio) < n_cheio:
79             self.stackedWidget.setCurrentIndex(2)
80     elif pagina == 3 and garrafasA != 0:
81         self.stackedWidget.setCurrentIndex(3)
82         self.GATotal.setText(str(garrafasA))
83         self.valor.setText(str(valor_garrafa))
84         resultado = decimal.Decimal(garrafasA)*decimal.Decimal(valor_garrafa)
85         resultado1 = round(resultado, 2)
86         self.retorno.setText(str(resultado1))
87     elif pagina == 3 and garrafasA == 0:
88         d.put(0)
89         pagina = 0
90     elif pagina == 4:
91         self.stackedWidget.setCurrentIndex(4)
92         time.sleep(2)
93         while cheio == 1:
94             self.stackedWidget.setCurrentIndex(5)
95             if flag_5 == 0:
96                 file = open("Email.txt", "w")
97                 lines = ["Contendor Cheio\n\n", "Número de garrafas no contendor: ", str(garrafasAT), "\n\nMorada: Penafiel, Porto \n4560"]
98                 file.writelines(lines)
99                 file.close()
100                 file = open("Email.txt", "r")
101                 msg = MIMEText(file.read())
102                 file.close()
103
104                 de = "reversevmachine@gmail.com"
105                 msg['Subject'] = assunto
106                 msg['From'] = de
107                 msg['To'] = para
```

Python file length: 153

```
ProcessPython_funcional_email.py x
107         msg['To'] = para
108         username = 'reversevmachine@gmail.com'
109         password = 'reverse2017'
110
111         s = smtplib.SMTP(host='smtp.gmail.com', port=587)
112         s.ehlo()
113         s.starttls()
114         s.login(username,password)
115         s.sendmail(de, [para], msg.as_string())
116         s.quit()
117         flag_5 = 1
118
119         pagina = d.get()
120         pagina = 0
121         d.put(pagina)
122     elif pagina == 6:
123         self.stackedWidget.setCurrentIndex(6)
124         self.GATotal_cheio.setText(str(garrafasA))
125         self.valor_cheio.setText(str(valor_garrafa))
126         resultado = decimal.Decimal(garrafasA)*decimal.Decimal(valor_garrafa)
127         resultado1 = round(resultado,2)
128         self.retorno_cheio.setText(str(resultado1))
129
130         self.iniciar_button.clicked.connect(self.teste)
131         self.terminar.clicked.connect(self.terminar_func)
132         self.iniciar_button_2.clicked.connect(self.teste)
133         self.talao.clicked.connect(self.talao_func)
134         self.talao_cheio.clicked.connect(self.talao_func)
135
136     def inicio(self):
137         pagina = 0
138         d.put(pagina)
139
140     def talao_func(self):
141         pagina = 4
142         d.put(pagina)
143         return
144
145     def terminar_func(self):
146         pagina = 3
147         d.put(pagina)
148         return
149
150     def teste(self):
151         global clicked
152         clicked = 1
153         return
154
155     def processo(self):
156         global clicked
157         clicked = 0
158         global garrafasA
159         garrafasA = 0
160         global garrafasR
```

Python file

```
ProcessPython_funcional_email.py
160     global garrafasR
161     garrafasR = 0
162     global garrafasAT
163     garrafasAT = 0
164     global garrafasRT
165     garrafasRT = 0
166     global w
167     w=0
168     global cheio
169     cheio = 0
170     while True:
171         if w==0:
172             pagina = 0
173             d.put(pagina)
174             w=1
175         if clicked == 1 and cheio == 0:
176             clicked = 0
177             v=0
178             if v==0:
179                 f=0
180                 v=1
181                 while f==0:
182                     GPIO.setwarnings(False)
183                     GPIO.setmode(GPIO.BOARD)
184                     entradaIR = 11
185                     GPIO.setup(entradaIR,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
186                     entrada = GPIO.input(entradaIR)
187                     if entrada==1:
188                         f=1
189             if entrada == 1:
190                 pagina = 1
191                 d.put(pagina)
192                 for z in range(1):
193                     x=0
194                     if x == 0:
195                         GPIO.setwarnings(False)
196                         GPIO.setmode(GPIO.BOARD)
197                         entradaIR = 11
198                         saidaIR = 13
199                         motor1A = 3
200                         motor1B = 5
201                         motor1E = 7
202                         motor2A = 21
203                         motor2B = 19
204                         motor2E = 15
205                         LEDs = 23
206                         Frequencia = 100
207                         DC = 100
208
209                         flag = 1
210                         flag1 = 1
211                         flag2 = 0
212                         flag3 = 0
213                         flag4 = 0
```

Python file

```
ProcessPython_funcional_email.py x
213         flag4 = 0
214
215         GPIO.setup(motor1A,GPIO.OUT)
216         GPIO.setup(motor1B,GPIO.OUT)
217         GPIO.setup(motor1E,GPIO.OUT)
218         GPIO.setup(motor2A,GPIO.OUT)
219         GPIO.setup(motor2B,GPIO.OUT)
220         GPIO.setup(motor2E,GPIO.OUT)
221         GPIO.setup(LEDs,GPIO.OUT)
222         GPIO.setup(entradaIR,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
223         GPIO.setup(saidaIR,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
224
225         pwm1=GPIO.PWM(motor1E,Frequencia)
226         pwm2=GPIO.PWM(motor2E,Frequencia)
227         entrada = GPIO.input(entradaIR)
228         saida = GPIO.input(saidaIR)
229         GPIO.output(LEDs,GPIO.LOW)
230         x=1
231
232         entrada = GPIO.input(entradaIR)
233         saida = GPIO.input(saidaIR)
234         while entrada == 0:
235             entrada = GPIO.input(entradaIR)
236             GPIO.output(motor1E,GPIO.LOW)
237
238         while ((entrada == 1 and saida == 0) or flag == 0):
239             flag=0
240             GPIO.output(motor2E,GPIO.LOW)
241             pwm2.stop()
242             pwm1.start(DC)
243             GPIO.output(motor1A,GPIO.HIGH)
244             GPIO.output(motor1B,GPIO.LOW)
245             GPIO.output(motor1E,GPIO.HIGH)
246             saida = GPIO.input(saidaIR)
247             if (saida == 1):
248                 flag=1
249                 flag1=0
250
251             if saida == 0:
252                 saida = GPIO.input(saidaIR)
253                 GPIO.output(motor2E,GPIO.LOW)
254
255         while ((entrada == 0 and saida == 1) or flag1 == 0):
256             GPIO.output(motor1E,GPIO.LOW)
257             pwm1.stop()
258             time.sleep(0.2)
259             GPIO.output(LEDs,GPIO.HIGH)
260             pwm2.start(DC)
261             GPIO.output(motor2A,GPIO.HIGH)
262             GPIO.output(motor2B,GPIO.LOW)
263             GPIO.output(motor2E,GPIO.HIGH)
264
265         camera = PiCamera()
266         camera.resolution = (640, 480)
```

Python file

```
ProcessPython_funcional_email.py [3]

266 camera.resolution = (640, 480)
267 camera.framerate = 32
268 rawCapture = PiRGBArray(camera, size=(640, 480))
269
270 symbol_cascade = cv2.CascadeClassifier('cascade_5_treino.xml')
271 Detected = 0
272 first_time = time.time()
273 time.sleep(0.1)
274 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
275     image = frame.array
276     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
277     symbol = symbol_cascade.detectMultiScale(gray, 1.3, 5)
278     tempo = time.time() - first_time
279     if tempo >= tempo_analise:
280         camera.close()
281         x=Detected
282         break
283     if Detected ==1:
284         camera.close()
285         break
286     if Detected == 0:
287         for (x,y,w,h) in symbol:
288             cv2.rectangle(image, (x,y), (x+w, y+h), (255,255,0), 2)
289             cv2.imshow("Frame", image)
290             Detected = 1
291             GPIO.output(motor2E,GPIO.LOW)
292             pwm2.stop()
293     if Detected == 1:
294         x=Detected
295         rawCapture.truncate(0)
296
297     if x==0:
298         GPIO.output(motor2E,GPIO.LOW)
299         pwm2.stop()
300         GPIO.output(LEDs,GPIO.LOW)
301         flag1 = 1
302         flag2 = 1
303     if(x==1):
304         flag1=1
305         GPIO.output(motor2E,GPIO.LOW)
306         pwm2.stop()
307         GPIO.output(LEDs,GPIO.LOW)
308
309 while flag1 == 1 and saida == 1 and x == 1:
310     if flag4 == 0:
311         garrafasA += 1
312         garrafasAT += 1
313         file = open("Garrafas.txt","w")
314         lines = [str(garrafasAT), " Garrafas Aceites", "\n", str(garrafasRT), " Garrafas Rejeitadas"]
315         file.writelines(lines)
316         file.close()
317         flag4 = 1
318     time.sleep(1.5)
319     saida = GPIO.input(saidaIR)
```

Python file

```

ProcessPython_funcional_email.py [3]
314         lines = [str(garrafasAI), " Garrafas Aceites", "\n", str(garrafasRT), " Garrafas Rejeitadas"]
315         file.writelines(lines)
316         file.close()
317         flag4 = 1
318         time.sleep(1.5)
319         saida = GPIO.input(saidaIR)
320         pwm1.start(DC)
321         GPIO.output(motor1A,GPIO.HIGH)
322         GPIO.output(motor1B,GPIO.LOW)
323         GPIO.output(motor1E,GPIO.HIGH)
324         saida = GPIO.input(saidaIR)
325         self.nGarrafasA.setText(str(garrafasA))
326         self.nGarrafasR.setText(str(garrafasR))
327         if saida == 0:
328             GPIO.output(motor1E,GPIO.LOW)
329             pwm1.stop()
330             clicked = 0
331             pagina = 2
332             d.put(pagina)
333
334     entrada = GPIO.input(entradaIR)
335     while flag2 == 1 and entrada == 0:
336         if flag3 == 0:
337             garrafasR += 1
338             garrafasRT += 1
339             file = open("Garrafas.txt","w")
340             lines = [str(garrafasAI), " Garrafas Aceites", "\n", str(garrafasRT), " Garrafas Rejeitadas"]
341             file.writelines(lines)
342             file.close()
343             flag3 = 1
344             time.sleep(0.5)
345             pwm1.start(DC)
346             GPIO.output(motor1A,GPIO.LOW)
347             GPIO.output(motor1B,GPIO.HIGH)
348             GPIO.output(motor1E,GPIO.HIGH)
349             entrada = GPIO.input(entradaIR)
350             self.nGarrafasA.setText(str(garrafasA))
351             self.nGarrafasR.setText(str(garrafasR))
352
353         while flag2 == 1 and entrada == 1:
354             GPIO.output(motor1E,GPIO.LOW)
355             pwm1.stop()
356             clicked = 0
357             pagina = 2
358             d.put(pagina)
359             flag2=0
360
361 if __name__ == "__main__":
362     app = QtWidgets.QApplication(sys.argv)
363     window = MyApp()
364     window.show()
365     sys.exit(app.exec_())
366

```



## Anexo F. Ambiente Gráfico

```
mainwindow.ui
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3      <class>MainWindow</class>
4      <widget class="QMainWindow" name="MainWindow">
5          <property name="enabled">
6              <bool>true</bool>
7          </property>
8          <property name="geometry">
9              <rect>
10                 <x>0</x>
11                 <y>0</y>
12                 <width>750</width>
13                 <height>517</height>
14             </rect>
15         </property>
16         <property name="windowTitle">
17             <string>MainWindow</string>
18         </property>
19         <widget class="QWidget" name="centralWidget">
20             <widget class="QStackedWidget" name="stackedWidget">
21                 <property name="geometry">
22                     <rect>
23                         <x>0</x>
24                         <y>0</y>
25                         <width>751</width>
26                         <height>511</height>
27                     </rect>
28                 </property>
29                 <property name="currentIndex">
30                     <number>5</number>
31                 </property>
32                 <widget class="QWidget" name="index0">
33                     <widget class="QLabel" name="label_22">
34                         <property name="geometry">
35                             <rect>
36                                 <x>0</x>
37                                 <y>-40</y>
38                                 <width>761</width>
39                                 <height>561</height>
40                             </rect>
41                         </property>
42                         <property name="text">
43                             <string/>
44                         </property>
45                         <property name="pixmap">
46                             <pixmap>fundo.jpg</pixmap>
47                         </property>
48                     </widget>
49                     <widget class="QLabel" name="label_3">
50                         <property name="geometry">
51                             <rect>
52                                 <x>280</x>
53                                 <y>60</y>
54                                 <width>211</width>
```

eXtensible Markup Language file

```
mainwindow.ui x
54      <width>211</width>
55      <height>191</height>
56    </rect>
57  </property>
58  <property name="text">
59    <string/>
60  </property>
61  <property name="pixmap">
62    <pixmap>bottle.jpg</pixmap>
63  </property>
64 </widget>
65 <widget class="QPushButton" name="iniciar_button">
66   <property name="geometry">
67     <rect>
68       <x>330</x>
69       <y>280</y>
70       <width>101</width>
71       <height>31</height>
72     </rect>
73   </property>
74   <property name="text">
75     <string>Iniciar</string>
76   </property>
77 </widget>
78 </widget>
79 <widget class="QWidget" name="index1">
80   <widget class="QLabel" name="label_23">
81     <property name="geometry">
82       <rect>
83         <x>0</x>
84         <y>-30</y>
85         <width>861</width>
86         <height>541</height>
87       </rect>
88     </property>
89     <property name="text">
90       <string/>
91     </property>
92     <property name="pixmap">
93       <pixmap>fundo.jpg</pixmap>
94     </property>
95   </widget>
96   <widget class="QLabel" name="label_4">
97     <property name="geometry">
98       <rect>
99         <x>270</x>
100        <y>100</y>
101        <width>201</width>
102        <height>181</height>
103      </rect>
104    </property>
105    <property name="text">
106      <string/>
107    </property>
```

eXtensible Markup Language file



```
mainwindow.ui x
107     </property>
108     <property name="pixmap">
109         <pixmap>process-icon.jpg</pixmap>
110     </property>
111 </widget>
112 </widget>
113 <widget class="QWidget" name="index2">
114     <widget class="QLabel" name="label_24">
115         <property name="geometry">
116             <rect>
117                 <x>0</x>
118                 <y>-40</y>
119                 <width>861</width>
120                 <height>541</height>
121             </rect>
122         </property>
123         <property name="text">
124             <string/>
125         </property>
126         <property name="pixmap">
127             <pixmap>fun.do.jpg</pixmap>
128         </property>
129     </widget>
130     <widget class="QLabel" name="label_5">
131         <property name="geometry">
132             <rect>
133                 <x>140</x>
134                 <y>150</y>
135                 <width>161</width>
136                 <height>21</height>
137             </rect>
138         </property>
139         <property name="text">
140             <string>Garrafas Rejeitadas:</string>
141         </property>
142         <property name="alignment">
143             <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
144         </property>
145     </widget>
146     <widget class="QLabel" name="label_6">
147         <property name="geometry">
148             <rect>
149                 <x>160</x>
150                 <y>100</y>
151                 <width>141</width>
152                 <height>21</height>
153             </rect>
154         </property>
155         <property name="text">
156             <string>Garrafas Aceites:</string>
157         </property>
158         <property name="alignment">
159             <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
160         </property>
```

eXtensible Markup Language file

```
mainwindow.ui
160     </property>
161 </widget>
162 <widget class="QLabel" name="nGarrafasA">
163     <property name="geometry">
164         <rect>
165             <x>320</x>
166             <y>100</y>
167             <width>68</width>
168             <height>21</height>
169         </rect>
170     </property>
171     <property name="frameShape">
172         <enum>QFrame::Box</enum>
173     </property>
174     <property name="text">
175         <string>TextLabel</string>
176     </property>
177     <property name="alignment">
178         <set>Qt::AlignCenter</set>
179     </property>
180 </widget>
181 <widget class="QLabel" name="nGarrafasR">
182     <property name="geometry">
183         <rect>
184             <x>320</x>
185             <y>150</y>
186             <width>68</width>
187             <height>21</height>
188         </rect>
189     </property>
190     <property name="frameShape">
191         <enum>QFrame::Box</enum>
192     </property>
193     <property name="text">
194         <string>TextLabel</string>
195     </property>
196     <property name="alignment">
197         <set>Qt::AlignCenter</set>
198     </property>
199 </widget>
200 <widget class="QPushButton" name="iniciar_button_2">
201     <property name="geometry">
202         <rect>
203             <x>180</x>
204             <y>220</y>
205             <width>101</width>
206             <height>31</height>
207         </rect>
208     </property>
209     <property name="text">
210         <string>Iniciar</string>
211     </property>
212 </widget>
213 <widget class="QPushButton" name="terminar">
```

eXtensible Markup Language file

```

mainwindow.ui x
213 <widget class="QPushButton" name="terminar">
214 <property name="geometry">
215 <rect>
216 <x>320</x>
217 <y>220</y>
218 <width>101</width>
219 <height>31</height>
220 </rect>
221 </property>
222 <property name="text">
223 <string>Terminar</string>
224 </property>
225 </widget>
226 <widget class="QLabel" name="label_29">
227 <property name="geometry">
228 <rect>
229 <x>420</x>
230 <y>60</y>
231 <width>191</width>
232 <height>211</height>
233 </rect>
234 </property>
235 <property name="text">
236 <string/>
237 </property>
238 <property name="pixmap">
239 <pixmap>pet_bottle.jpg</pixmap>
240 </property>
241 </widget>
242 </widget>
243 <widget class="QWidget" name="index3">
244 <widget class="QLabel" name="label_25">
245 <property name="geometry">
246 <rect>
247 <x>0</x>
248 <y>-20</y>
249 <width>861</width>
250 <height>541</height>
251 </rect>
252 </property>
253 <property name="text">
254 <string/>
255 </property>
256 <property name="pixmap">
257 <pixmap>fundo.jpg</pixmap>
258 </property>
259 </widget>
260 <widget class="QLabel" name="label_7">
261 <property name="geometry">
262 <rect>
263 <x>130</x>
264 <y>140</y>
265 <width>121</width>
266 <height>21</height>

```

eXtensible Markup Language file



```

mainwindow.ui x
266         <height>21</height>
267     </rect>
268 </property>
269 <property name="text">
270     <string>Garrafas Aceites:</string>
271 </property>
272 <property name="alignment">
273     <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
274 </property>
275 </widget>
276 <widget class="QLabel" name="label_9">
277     <property name="geometry">
278         <rect>
279             <x>310</x>
280             <y>140</y>
281             <width>121</width>
282             <height>21</height>
283         </rect>
284     </property>
285     <property name="text">
286         <string>Total a Receber:</string>
287     </property>
288     <property name="alignment">
289         <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
290     </property>
291 </widget>
292 <widget class="QLabel" name="GATotal">
293     <property name="geometry">
294         <rect>
295             <x>180</x>
296             <y>170</y>
297             <width>68</width>
298             <height>21</height>
299         </rect>
300     </property>
301     <property name="frameShape">
302         <enum>QFrame::Box</enum>
303     </property>
304     <property name="text">
305         <string>TextLabel</string>
306     </property>
307     <property name="alignment">
308         <set>Qt::AlignCenter</set>
309     </property>
310 </widget>
311 <widget class="QLabel" name="label_8">
312     <property name="geometry">
313         <rect>
314             <x>260</x>
315             <y>170</y>
316             <width>16</width>
317             <height>21</height>
318         </rect>
319     </property>

```

eXtensible Markup Language file

```

mainwindow.ui x
319     </property>
320     <property name="text">
321         <string>X</string>
322     </property>
323 </widget>
324 <widget class="QLabel" name="valor">
325     <property name="geometry">
326         <rect>
327             <x>270</x>
328             <y>170</y>
329             <width>31</width>
330             <height>21</height>
331         </rect>
332     </property>
333     <property name="text">
334         <string>TextLabel</string>
335     </property>
336 </widget>
337 <widget class="QLabel" name="label_20">
338     <property name="geometry">
339         <rect>
340             <x>300</x>
341             <y>170</y>
342             <width>16</width>
343             <height>21</height>
344         </rect>
345     </property>
346     <property name="text">
347         <string>=</string>
348     </property>
349 </widget>
350 <widget class="QLabel" name="retorno">
351     <property name="geometry">
352         <rect>
353             <x>320</x>
354             <y>170</y>
355             <width>68</width>
356             <height>21</height>
357         </rect>
358     </property>
359     <property name="frameShape">
360         <enum>QFrame::Box</enum>
361     </property>
362     <property name="text">
363         <string>TextLabel</string>
364     </property>
365     <property name="alignment">
366         <set>Qt::AlignCenter</set>
367     </property>
368 </widget>
369 <widget class="QLabel" name="label_18">
370     <property name="geometry">
371         <rect>
372             <x>390</x>

```

eXtensible Markup Language file

```
mainwindow.ui X
372         <x>390</x>
373         <y>170</y>
374         <width>16</width>
375         <height>21</height>
376     </rect>
377 </property>
378 <property name="text">
379     <string>€</string>
380 </property>
381 </widget>
382 <widget class="QPushButton" name="talao">
383     <property name="geometry">
384         <rect>
385             <x>220</x>
386             <y>240</y>
387             <width>111</width>
388             <height>31</height>
389         </rect>
390     </property>
391     <property name="text">
392         <string>Imprimir Talão</string>
393     </property>
394 </widget>
395 <widget class="QLabel" name="label_30">
396     <property name="geometry">
397         <rect>
398             <x>440</x>
399             <y>50</y>
400             <width>201</width>
401             <height>251</height>
402         </rect>
403     </property>
404     <property name="text">
405         <string/>
406     </property>
407     <property name="pixmap">
408         <pixmap>coupon.jpg</pixmap>
409     </property>
410 </widget>
411 </widget>
412 <widget class="QWidget" name="index4">
413     <widget class="QLabel" name="label_26">
414         <property name="geometry">
415             <rect>
416                 <x>0</x>
417                 <y>-20</y>
418                 <width>861</width>
419                 <height>541</height>
420             </rect>
421         </property>
422         <property name="text">
423             <string/>
424         </property>
425         <property name="pixmap">
```

eXtensible Markup Language file



```
mainwindow.ui x
425 <property name="pixmap">
426 <pixmap>funido.jpg</pixmap>
427 </property>
428 </widget>
429 <widget class="QLabel" name="label">
430 <property name="geometry">
431 <rect>
432 <x>250</x>
433 <y>70</y>
434 <width>231</width>
435 <height>51</height>
436 </rect>
437 </property>
438 <property name="font">
439 <font>
440 <pointsize>20</pointsize>
441 <weight>75</weight>
442 <bold>true</bold>
443 </font>
444 </property>
445 <property name="text">
446 <string>A Imprimir Talão</string>
447 </property>
448 <property name="alignment">
449 <set>Qt::AlignCenter</set>
450 </property>
451 </widget>
452 <widget class="QLabel" name="label_10">
453 <property name="geometry">
454 <rect>
455 <x>250</x>
456 <y>120</y>
457 <width>261</width>
458 <height>261</height>
459 </rect>
460 </property>
461 <property name="text">
462 <string/>
463 </property>
464 <property name="pixmap">
465 <pixmap>print-flat.jpg</pixmap>
466 </property>
467 </widget>
468 </widget>
469 <widget class="QWidget" name="index5">
470 <widget class="QLabel" name="label_27">
471 <property name="geometry">
472 <rect>
473 <x>0</x>
474 <y>-20</y>
475 <width>861</width>
476 <height>541</height>
477 </rect>
478 </property>
```

eXtensible Markup Language file

```

mainwindow.ui
478     </property>
479     <property name="text">
480         <string/>
481     </property>
482     <property name="pixmap">
483         <pixmap>funido.jpg</pixmap>
484     </property>
485 </widget>
486 <widget class="QLabel" name="label_11">
487     <property name="geometry">
488         <rect>
489             <x>190</x>
490             <y>70</y>
491             <width>351</width>
492             <height>41</height>
493         </rect>
494     </property>
495     <property name="font">
496         <font>
497             <pointsize>20</pointsize>
498             <weight>75</weight>
499             <bold>true</bold>
500         </font>
501     </property>
502     <property name="text">
503         <string>MANUTENÇÃO</string>
504     </property>
505     <property name="alignment">
506         <set>Qt::AlignCenter</set>
507     </property>
508 </widget>
509 <widget class="QLabel" name="label_2">
510     <property name="geometry">
511         <rect>
512             <x>280</x>
513             <y>130</y>
514             <width>201</width>
515             <height>211</height>
516         </rect>
517     </property>
518     <property name="text">
519         <string/>
520     </property>
521     <property name="pixmap">
522         <pixmap>maintenance-icon.jpg</pixmap>
523     </property>
524 </widget>
525 </widget>
526 <widget class="QWidget" name="index6">
527     <widget class="QLabel" name="label_28">
528         <property name="geometry">
529             <rect>
530                 <x>0</x>
531                 <y>-20</y>

```

eXtensible Markup Language file



```

mainwindow.ui
531         <y>-20</y>
532         <width>861</width>
533         <height>541</height>
534     </rect>
535 </property>
536 <property name="text">
537     <string/>
538 </property>
539 <property name="pixmap">
540     <pixmap>fun.do.jpg</pixmap>
541 </property>
542 </widget>
543 <widget class="QLabel" name="label_15">
544     <property name="geometry">
545         <rect>
546             <x>370</x>
547             <y>20</y>
548             <width>201</width>
549             <height>211</height>
550         </rect>
551     </property>
552     <property name="text">
553         <string/>
554     </property>
555     <property name="pixmap">
556         <pixmap>garbage_full.jpg</pixmap>
557     </property>
558 </widget>
559 <widget class="QLabel" name="label_16">
560     <property name="geometry">
561         <rect>
562             <x>140</x>
563             <y>70</y>
564             <width>241</width>
565             <height>41</height>
566         </rect>
567     </property>
568     <property name="font">
569         <font>
570             <pointsize>20</pointsize>
571             <weight>75</weight>
572             <bold>true</bold>
573         </font>
574     </property>
575     <property name="text">
576         <string>CONTENTOR CHEIO</string>
577     </property>
578 </widget>
579 <widget class="QLabel" name="label_17">
580     <property name="geometry">
581         <rect>
582             <x>140</x>
583             <y>120</y>
584             <width>241</width>

```

Qtensible Markup Language file

```

mainwindow.ui
584         <width>241</width>
585         <height>21</height>
586     </rect>
587 </property>
588 <property name="text">
589     <string>Pedimos desculpa pelo incómodo</string>
590 </property>
591 </widget>
592 <widget class="QLabel" name="label_13">
593     <property name="geometry">
594         <rect>
595             <x>210</x>
596             <y>240</y>
597             <width>121</width>
598             <height>21</height>
599         </rect>
600     </property>
601     <property name="text">
602         <string>Garrafas Aceites:</string>
603     </property>
604     <property name="alignment">
605         <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
606     </property>
607 </widget>
608 <widget class="QLabel" name="label_12">
609     <property name="geometry">
610         <rect>
611             <x>390</x>
612             <y>240</y>
613             <width>121</width>
614             <height>21</height>
615         </rect>
616     </property>
617     <property name="text">
618         <string>Total a receber:</string>
619     </property>
620 </widget>
621 <widget class="QLabel" name="GATotal_cheio">
622     <property name="geometry">
623         <rect>
624             <x>260</x>
625             <y>280</y>
626             <width>68</width>
627             <height>21</height>
628         </rect>
629     </property>
630     <property name="frameShape">
631         <enum>QFrame::Box</enum>
632     </property>
633     <property name="text">
634         <string>TextLabel</string>
635     </property>
636     <property name="alignment">
637         <set>Qt::AlignCenter</set>

```

eXtensible Markup Language file

```
mainwindow.ui x
637         <set>Qt::AlignCenter</set>
638     </property>
639 </widget>
640 <widget class="QLabel" name="label_14">
641     <property name="geometry">
642         <rect>
643             <x>330</x>
644             <y>280</y>
645             <width>16</width>
646             <height>21</height>
647         </rect>
648     </property>
649     <property name="text">
650         <string>X</string>
651     </property>
652 </widget>
653 <widget class="QLabel" name="valor_cheio">
654     <property name="geometry">
655         <rect>
656             <x>340</x>
657             <y>280</y>
658             <width>31</width>
659             <height>21</height>
660         </rect>
661     </property>
662     <property name="text">
663         <string>TextLabel</string>
664     </property>
665 </widget>
666 <widget class="QLabel" name="label_21">
667     <property name="geometry">
668         <rect>
669             <x>370</x>
670             <y>280</y>
671             <width>16</width>
672             <height>21</height>
673         </rect>
674     </property>
675     <property name="text">
676         <string>=</string>
677     </property>
678 </widget>
679 <widget class="QLabel" name="retorno_cheio">
680     <property name="geometry">
681         <rect>
682             <x>390</x>
683             <y>280</y>
684             <width>68</width>
685             <height>21</height>
686         </rect>
687     </property>
688     <property name="frameShape">
689         <enum>QFrame::Box</enum>
690     </property>
```

eXtensible Markup Language file



```
mainwindow.ui x
691 <property name="text">
692   <string>TextLabel</string>
693 </property>
694 <property name="alignment">
695   <set>Qt::AlignCenter</set>
696 </property>
697 </widget>
698 <widget class="QLabel" name="label_19">
699   <property name="geometry">
700     <rect>
701       <x>460</x>
702       <y>280</y>
703       <width>16</width>
704       <height>21</height>
705     </rect>
706   </property>
707   <property name="text">
708     <string>€</string>
709   </property>
710 </widget>
711 <widget class="QPushButton" name="talao_cheio">
712   <property name="geometry">
713     <rect>
714       <x>300</x>
715       <y>340</y>
716       <width>111</width>
717       <height>31</height>
718     </rect>
719   </property>
720   <property name="text">
721     <string>Imprimir Talão</string>
722   </property>
723 </widget>
724 </widget>
725 </widget>
726 </widget>
727 <widget class="QMenuBar" name="menuBar">
728   <property name="geometry">
729     <rect>
730       <x>0</x>
731       <y>0</y>
732       <width>750</width>
733       <height>27</height>
734     </rect>
735   </property>
736 </widget>
737 <widget class="QStatusBar" name="statusBar"/>
738 </widget>
739 <layoutdefault spacing="6" margin="11"/>
740 <resources/>
741 <connections/>
742 </ui>
743
```

eXtensible Markup Language file